EMV®
Secure Remote Commerce

Specification – API

Version 1.2

June 2021

Legal Notice

The EMV® Specifications are provided “AS IS” without warranties of any kind, and EMVCo neither assumes nor accepts any liability for any errors or omissions contained in these Specifications. 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 THESE SPECIFICATIONS.

EMVCo makes no representations or warranties with respect to intellectual property rights of any third parties in or in relation to the Specifications. EMVCo undertakes no responsibility to determine whether any implementation of the EMV® Specifications 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 the EMV® Specifications should consult an intellectual property attorney before any such implementation.

Without limiting the foregoing, the Specifications 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 these Specifications 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 the EMV® Specifications.

Revision Log – Version 1.2

The following changes have been made to the document since the publication of version 1.1:

- Editorial changes throughout the document
- Introduction of the concept of the SRC Specifications, encompassing the suite of SRC documents (Section 1.4.2 Published EMVCo Documents)
- Addition of the following subsections in Section 2.1 Complex Data Objects and the renumbering of existing subsections to accommodate them:
  - 2.1.1 AcceptanceChannelRelatedData,
  - 2.1.2 AcceptanceChannelData
  - 2.1.3 AdditionalAmount
  - 2.1.18 DeliveryContactDetails
  - 2.1.21 DigitalCardUpdateNotification
  - 2.1.25 EnrollmentReferenceData
  - 2.1.40 VerificationData
- Deprecation of Section 2.1.11 ConfirmationData and replacement with Section 2.1.12 ConfirmationData2
- Addition of new individual attributes in Table 2.42
- Addition of new enumerations in Table 2.43
- Replacement of masking rules in Section 2.5 Masking Rule with a reference to the masking rules in the SRC Core Specification
- Removal of the URI format (Section 5.1.1 URI Format)
- Introduction of a new Section 5.1.5 Recognition
- Changes to the descriptions of the API operations in Section 5 Server-Side API to bring consistency across the SRC Specifications
- Edits to individual APIs in Section 5 Server-Side API
- Introduction of new operations:
  - Section 5.2.4 Get Card Data
  - Section 5.5.4 Make Payment
- Introduction of new Section 6 Notification Service
- Introduction of new Annex A EMVCo Specification Mapping

Contents

Legal Notice ..... **i**

Revision Log – Version 1.2 ..... **ii**

Contents ..... **iii**

Tables ..... **vii**

1 Introduction **1**

1.1 Scope 1
1.2 Constraints 1
1.3 Audience 1
1.4 References 2
1.4.1 Normative References 2
1.4.2 Published EMVCo Documents 2
1.5 Definitions 3
1.6 Notational Conventions 3
1.6.1 Abbreviations 3
1.6.2 Terminology and Conventions 4

2 Data Dictionary **5**

2.1 Complex Data Objects 5
2.1.1 AcceptanceChannelRelatedData 5
2.1.2 AcceptanceChannelData 6
2.1.3 AdditionalAmount 6
2.1.4 Address 7
2.1.5 AppInstance 8
2.1.6 AssuranceData 8
2.1.7 Card 15
2.1.8 CardholderData 17
2.1.9 CommunicationsConsent 18
2.1.10 ComplianceSettings 18
2.1.11 ~~**ConfirmationData**~~ **DEPRECATED** 19
2.1.12 ConfirmationData2 21
2.1.13 Consent 23
2.1.14 Consumer 24
2.1.15 ConsumerIdentity 25
2.1.16 Dcf 25
2.1.17 DeviceData 26
2.1.18 DeliveryContactDetails 26
2.1.19 DigitalCardData 27

2.1.20 DigitalCardFeature 28
2.1.21 DigitalCardUpdateNotification 29
2.1.22 DpaData 30
2.1.23 DpaTransactionOptions 31
2.1.24 DynamicData 35
2.1.25 EnrollmentReferenceData 35
2.1.26 Error 36
2.1.27 ErrorDetail 36
2.1.28 EventHistory 37
2.1.29 IdentityValidationChannel 38
2.1.30 MaskedAddress 39
2.1.31 MaskedCard 40
2.1.32 MaskedConsumer 43
2.1.33 MaskedConsumerIdentity 45
2.1.34 Payload 46
2.1.35 PaymentOptions 52
2.1.36 PaymentToken 52
2.1.37 PhoneNumber 53
2.1.38 SrcProfile 53
2.1.39 TransactionAmount 54
2.1.40 VerificationData 55
2.2 Individual Attributes 60
2.3 Enumerations 64
2.4 Signed Checkout Objects 66
2.4.1 Checkout Request JWS 66
2.4.2 Checkout Payload Response 69
2.4.3 JWS JOSE Header 74
2.5 Masking Rule 74
3 Federated Identity 75
3.1 Authorisation Token 75
3.1.1 Token Header 75
3.1.2 Token Claims 76
3.1.3 Notes on Authentication 80
4 SRCI – DCF Interaction 82
4.1 Interaction Mechanisms 82
4.2 Launch The DCF 83
4.3 Redirect back to SRCI 83

5 Server-Side API 86
5.1 API Principles 86
5.1.1 Common HTTP Status Codes 86
5.1.2 Error Handling 87
5.1.3 Conditionality of Data 87
5.1.4 Authorisation 87
5.1.5 Recognition 87
5.1.6 API Access Control 88
5.2 Card Service 88
5.2.1 Card Enrolment 88
5.2.2 Delete Card 91
5.2.3 Add Billing Address 93
5.2.4 Get Card Data 94
5.3 Address Service 95
5.3.1 Add Shipping Address 95
5.3.2 Delete Shipping Address 96
5.4 SRC Profile Service 98
5.4.1 Prepare SRC Profile 98
5.4.2 Add Consumer Identities 99
5.4.3 Unbind App Instance 101
5.5 Checkout Service 103
5.5.1 Prepare Checkout Data 103
5.5.2 Checkout 105
5.5.3 Get Payload 107
5.5.4 Make Payment 109
5.6 Confirmation Service 110
5.6.1 Confirmation 110
5.7 Identity Service 111
5.7.1 Identity Lookup 111
5.7.2 Initiate Identity Validation 112
5.7.3 Complete Identity Validation 113
5.7.4 Is Recognized 114
5.8 Public Keys Retrieval Service 115
5.8.1 Public Key Retrieval 116
6 Notification Service 118
6.1 Notifications Principles 118
6.1.1 Data Delivery Modes 118
6.1.2 Standard HTTP Status Codes 118

6.2 Card Update Event Notification 119
6.3 Identity Validation Completion Event Notification 120
6.4 Payment Notification 121

Annex A EMVCo Specification Mapping 123

A.1 Merchant-Presented Mode – QR Code Payload 123
A.1.1 SRC Data Elements 123
A.1.2 QR Code specific Data Elements for Seller Data 124
A.1.3 QR Code specific Data Elements for Consumer Data 125
A.1.4 QR Code Specific Data Elements for Additional Amounts 129

Tables

Table 1.1: Normative References 2
Table 1.2: EMVCo References 2
Table 2.1: AcceptanceChannelRelatedData 5
Table 2.2: AcceptanceChannelData 6
Table 2.3: AdditionalAmount 6
Table 2.4: Address 7
Table 2.5: AppInstance 8
Table 2.6: AssuranceData 8
Table 2.7: Card 15
Table 2.8: CardholderData 17
Table 2.9: CommunicationsConsent 18
Table 2.10: ComplianceSettings 18
Table 2.11: ~~ConfirmationData~~ **DEPRECATED** 19
Table 2.12: ConfirmationData2 21
Table 2.13: Consent 23
Table 2.14: Consumer 24
Table 2.15: ConsumerIdentity 25
Table 2.16: Dcf 25
Table 2.17: DeviceData 26
Table 2.18: DeliveryContactDetails 26
Table 2.19: DigitalCardData 27
Table 2.20: DigitalCardFeature 28
Table 2.21: DigitalCardUpdateNotification 29
Table 2.22: DpaData 30
Table 2.23: DpaTransactionOptions 31
Table 2.24: DynamicData 35
Table 2.25: EnrollmentReferenceData 35
Table 2.26: Error 36
Table 2.27: ErrorDetail 36
Table 2.28: EventHistory 37
Table 2.29: IdentityValidationChannel 38
Table 2.30: MaskedAddress 39
Table 2.31: MaskedCard 40
Table 2.32: MaskedConsumer 43
Table 2.33: MaskedConsumerIdentity 45
Table 2.34: Payload 46
Table 2.35: PaymentOptions 52
Table 2.36: PaymentToken 52
Table 2.37: PhoneNumber 53
Table 2.38: SrcProfile 53

Table 2.39: TransactionAmount 54
Table 2.40: VerificationData 55
Table 2.41: VerificationData Values 56
Table 2.42: Individual Attributes 60
Table 2.43: Enumerations 64
Table 2.44: Checkout Request JOSE Header 67
Table 2.45: Checkout Request Claim Set 67
Table 2.46: Checkout Payload Response 70
Table 2.47: JWS JOSE Header 74
Table 3.1: JOSE Header 75
Table 3.2: Federated ID Token Claim Set 76
Table 4.1: Error Codes 84
Table 5.1: Card Enrolment Definition (HTTP with JSON) 88
Table 5.2: Delete Card Definition (HTTP with JSON) 91
Table 5.3: Add Billing Address Definition (HTTP with JSON) 93
Table 5.4: Get Card Data Definition (HTTP with JSON) 94
Table 5.5: Add Shipping Address Definition (HTTP with JSON) 95
Table 5.6: Delete Shipping Address Definition (HTTP with JSON) 96
Table 5.7: Prepare SRC Profile Definition (HTTP with JSON) 98
Table 5.8: Add Consumer Identities Definition (HTTP with JSON) 100
Table 5.9: Unbind App Instance Definition (HTTP with JSON) 101
Table 5.10: Prepare Checkout Data Definition (HTTP with JSON) 103
Table 5.11: Checkout Definition (HTTP with JSON) 105
Table 5.12: Get Payload Definition (HTTP with JSON) 107
Table 5.13: Make Payment Definition (HTTP with JSON) 109
Table 5.14: Confirmation Definition (HTTP with JSON) 110
Table 5.15: Identity Lookup Definition (HTTP with JSON) 111
Table 5.16: Initiate Identity Validation Definition (HTTP with JSON) 112
Table 5.17: Complete Identity Validation Definition (HTTP with JSON) 113
Table 5.18: Is Recognized Definition (HTTP with JSON) 114
Table 5.19: Public Key Retrieval Definition (HTTP with JSON) 116
Table 6.1: Standard HTTP Status Codes 119
Table 6.2: Card Update Notification Definition (HTTP with JSON) 120
Table 6.3: Complete Identity Validation Notification Definition (HTTP with JSON) 120
Table 6.4: Payment Notification Definition (HTTP with JSON) 121
Table A.1: SRC API Usage for QR Code Payload 125
Table A.2: SRC API Usage for Bill Number 125
Table A.3: SRC API Usage for Mobile Number 126
Table A.4: SRC API Usage for Store Label 126
Table A.5: SRC API Usage for Loyalty Number 126
Table A.6: SRC API Usage for Reference Label 127
Table A.7: SRC API Usage for Customer Label 127

Table A.8: SRC API Usage for Terminal Label 127
Table A.9: SRC API Usage for Purpose of Transaction 128
Table A.10: SRC API Usage for Email 128
Table A.11: SRC API Usage for Phone Number 129
Table A.12: SRC API Usage for Address 129
Table A.13: SRC API Usage for Tip 130
Table A.14: SRC API Usage for Convenience Fee 130
Table A.15: SRC API Usage for Sub Total 131

1 Introduction

Secure Remote Commerce (SRC) is an evolution of remote commerce that provides for secure and interoperable card acceptance established through a standard specification.

This document, the EMV Secure Remote Commerce Specification – API, (hereafter the "SRC API Specification"), contains server-based APIs which can be used to securely build interfaces between SRC Systems and SRC System Participants. It is intended to be used in conjunction with the SRC Specifications (see Section 1.4.2 Published EMVCo Documents).

1.1 Scope

The SRC API Specification describes APIs to be used for the transmission of data between SRC Systems and SRC System Participants. These APIs are based on the following assumptions:

- The server-based APIs provide a toolkit for SRC System Participants
- They are not intended to provide context for all scenarios or use cases, and individual SRC Systems are responsible for creating implementation instructions for their SRC System Participants
- They do not preclude an SRC System from providing additional technical components to support their implementations
- The EMV SRC API specification offers levels of optionality for implementers of the specifications to add security layers based on the SRC solution provider's own security requirements and risk controls

1.2 Constraints

The SRC API Specification is designed to work within the constraints described in the SRC Core Specification. In particular, the SRC API Specification or any implementation of the SRC API Specification is not intended to replace or interfere with any international, regional, national or local laws and regulations; those governing requirements supersede any industry standards.

1.3 Audience

This document is intended for use by SRC Systems and SRC System Participants.

1.4 References

The latest version of any reference, including all published amendments, shall apply unless a publication date is explicitly stated.

1.4.1 Normative References

The standards in Table 1.1 may be associated with the SRC API Specification.

Table 1.1: Normative References

Reference	Publication Name
ISO 3166	Country Codes — ISO 3166
ISO 4217	Currency Codes — ISO 4217
ISO/IEC 7812	Identification cards — Identification of issuers
RFC 3447	Public-Key Cryptography Standards (https://tools.ietf.org/html/rfc3447)
RFC 7515	JSON Web Signature (https://tools.ietf.org/html/rfc7515)
RFC 7516	JSON Web Encryption (https://tools.ietf.org/html/rfc7516)
RFC 7517	JSON Web Key (https://tools.ietf.org/html/rfc7517)
RFC 7518	JSON Web Algorithms (https://tools.ietf.org/html/rfc7518)
RFC 7519	JSON Web Token (https://tools.ietf.org/html/rfc7519)

1.4.2 Published EMVCo Documents

The documents in Table 1.2 are related to or are associated with SRC and are located at www.emvco.com.

Table 1.2: EMVCo References

Reference	Publication Name
SRC Core Specification	EMV® Secure Remote Commerce Specification

Reference	Publication Name
SRC Reproduction Requirements	EMV® Secure Remote Commerce (SRC): Click to Pay Icon Reproduction Requirements
SRC UI Guidelines and Requirements	EMV® Secure Remote Commerce Specification – User Interface Guidelines and Requirements
SRC JavaScript SDK	EMV® Secure Remote Commerce Specification – JavaScript SDK
SRC Data Dictionary	EMV® Secure Remote Commerce Data Dictionary
SRC Version Management	EMV® Secure Remote Commerce Version Management for SRC API and SRC JavaScript SDK Specifications

Collectively, the term SRC Specifications refers to:

- SRC Core Specification
- SRC Reproduction Requirements
- SRC UI Guidelines and Requirements
- SRC API (this document)
- SRC JavaScript SDK
- SRC Data Dictionary
- SRC Version Management

1.5 Definitions

For the definition of the terms used in the SRC API Specification, refer to Table 1.3: Definitions in the SRC Core Specification. For definitions of data elements refer to the SRC API or the SRC Data Dictionary.

1.6 Notational Conventions

1.6.1 Abbreviations

For the definition of the abbreviations used in the SRC API Specification, refer to Section 1.9.1 Abbreviations in the SRC Core Specification.

1.6.2 Terminology and Conventions

For the definition of the terminology and conventions used in the SRC API Specification, refer to Section 1.9.2 Terminology and Conventions in the SRC Core Specification.

2 Data Dictionary

2.1 Complex Data Objects

Table 2.1 to Table 2.40 introduce the common data objects used across the API defined in the SRC API Specification. Each table defines a single data object.

The column headed R/C/O in each table refers to whether the data element is required, conditional or optional. The following notation is used:

- R = Required – always present
- C = Conditional – present under certain conditions (as specified in the description)
- O = Optional – can be present

2.1.1 AcceptanceChannelRelatedData

Table 2.1: AcceptanceChannelRelatedData

Data Element	R/C/O	Constraints	Description
acceptanceChannelType Type: AcceptanceChannelType	R	See AcceptanceChannelType	Type of acceptance channel
acceptanceChannelTechnology Type: AcceptanceChannelTechnology	O	See AcceptanceChannelTechnology	Technology used to transmit/receive the acceptance channel data
acceptanceChannelData Type: AcceptanceChannelData	R	See AcceptanceChannelData	Acceptance channel data

2.1.2 AcceptanceChannelData

Table 2.2: AcceptanceChannelData

Data Element	R/C/O	Constraints	Description
consumerData Type: JSON Object	C	Acceptance channel specific	Consumer supplied data, either manually entered (or supplied by other means, e.g. voice, camera etc.) or previously stored **Conditionality**: At least one of `consumerData` or `sellerData` is required
sellerData Type: JSON Object	C	Acceptance channel specific	Seller supplied data supplied over the acceptance channel technology, or other means **Conditionality**: At least one of `consumerData` or `sellerData` is required

2.1.3 AdditionalAmount

Table 2.3: AdditionalAmount

Data Element	R/C/O	Constraints	Description
additionalAmountType Type: AdditionalAmountType	R	See AdditionalAmount Type	Type of additional amount
additionalAmountValue Type: String	R		Value of the additional amount

2.1.4 Address

Table 2.4: Address

Data Element	R/C/O	Constraints	Description
addressId Type: String	O	UUID	Reference identifier of the address
name Type: String	O	Max Length = 100	Name of the ordering customer
line1 Type: String	O	Max Length = 75	Address line 1
line2 Type: String	O	Max Length = 75	Address line 2
line3 Type: String	O	Max Length = 75	Address line 3
city Type: String	O	Max Length = 50	Address city
state Type: String	O	Max Length = 30	Address state
countryCode Type: String	O	ISO 3166-1 alpha-2 country code	Address country code
zip Type: String	O	Max Length = 16	Address zip/postal code
deliveryContactDetails Type: DeliveryContactDetails	O	See DeliveryContactDetails	Delivery contact details for a shipping address
createTime Type: String (Numeric)	O	UTC time in Unix epoch format	Date and time the address was created
lastUsedTime Type: String (Numeric)	O	UTC time in Unix epoch format	Date and time the address was last used

2.1.5 AppInstance

Table 2.5: AppInstance

Data Element	R/C/O	Constraints	Description
userAgent Type: String	C	N/A	User agent string of the connecting client application **Conditionality**: • Required for browsers • Optional for non-browsers
applicationName Type: String	O	Max Length = 255	Name of the connecting client application
countryCode Type: String	O	ISO 3166-1 alpha-2 country code	The country where the Consumer is accessing the service from
deviceData Type: DeviceData	O	See DeviceData	Device specific data

2.1.6 AssuranceData

Table 2.6: AssuranceData

Data Element	R/C/O	Constraints	Description
verificationData Type: List<VerificationData>	R	See VerificationData	Set of verification data structures relating to different types of assurance

Data Element	R/C/O	Constraints	Description
~~**cardVerificationEntity**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	Length = 2	Entity performing card verification. Valid values are: • 01 SRC Initiator • 02 SRC System • 03 SRCPI • 04 DCF • 05 DPA • 06 - 99 Others
~~**cardVerificationMethod**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	Length = 2	Card verification check to validate that the PAN is active and valid at the Card Issuer. Valid values are: • 01 $0 authorisation, or single unit of currency authorisation • 02 Card Verification Number validation • 03 Postal code and address verification, where supported • 04 - 20 EMVCo future use • 21 - 99 SRC System specific
~~**cardVerificationResults**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	Length = 2	Verification status of the PAN. Valid values are: • 01 Verified • 02 Not Verified • 03 Not performed • 04 - 20 EMVCo future use • 21 - 99 SRC System specific

Data Element	R/C/O	Constraints	Description
~~**cardVerificationTimestamp**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	UTC time in Unix epoch format	Date and time when the card verification was conducted
~~**cardAssuranceData**~~ ~~Type: String~~ DEPRECATED	O		Data collected that is associated with the PAN and presented to the SRC System
~~**cardholderAuthenticationEntity**~~ ~~Type: String~~ DEPRECATED	O	Max Length = 64	Entity performing Cardholder authentication
~~**cardholderAuthenticationMethod**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	Length = 2	Card Issuer verification of the Cardholder. Valid values are: • 01 Use of a 3-D Secure ACS • 02 Mobile banking verification of the Cardholder with an authentication code • 03 Federated login systems • 04 A shared secret between the Card Issuer and the Cardholder such as One Time Passcode (OTP), activation code • 05 - 20 EMVCo future use • 21 - 99 SRC System specific

Data Element	R/C/O	Constraints	Description
~~**cardholderAuthenticationResults**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	Length = 2	Indicates whether the Cardholder was verified or not, and what the results are when verified. • 01 Verified • 02 Not Verified • 03 Not performed • 04 - 20 EMVCo future use • 21 - 99 SRC System specific
~~**cardholderAuthenticationTimestamp**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	UTC time in Unix epoch format	Date and time when the Cardholder authentication was conducted
~~**cardholderAssuranceData**~~ ~~Type: String~~ DEPRECATED	O		Data collected that is associated with the Cardholder and presented to the SRC System
~~**consumerVerificationEntity**~~ ~~Type: String~~ DEPRECATED	O	Max Length = 64	Entity performing Consumer verification

Data Element	R/C/O	Constraints	Description
~~**consumerVerificationMethod**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	Length = 2	The verification method used to verify Consumer credential. Valid values are: • 01 Static Passcode • 02 SMS One Time Passcode (OTP) • 03 Keyfob or EMV cardreader One Time Passcode (OTP) • 04 Application One Time Passcode (OTP) • 05 One Time Passcode (OTP) Other • 06 Knowledge Based Authentication (KBA) • 07 Out of Band Biometrics • 08 Out of Band Login • 09 Out of Band Other • 10 Risk-Based • 11 Other • 12 - 99 EMVCo future use
~~**consumerVerificationResults**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	Length = 2	Indicates whether the Consumer was verified or not, and what the results are when verified. Valid values are: • 01 Verified • 02 Not Verified • 03 Not performed • 04 - 20 EMVCo future use • 21 - 99 SRC System specific

Data Element	R/C/O	Constraints	Description
~~**consumerVerificationTimestamp**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	UTC time in Unix epoch format	Date and time when the Consumer verification was conducted
~~**consumerAssuranceData**~~ ~~Type: String~~ DEPRECATED	O		Data collected that is associated with the Consumer for assurance purposes
~~**deviceVerificationEntity**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	Length = 2	Entity performing device verification. The valid values are: • 01 SRC Initiator • 02 SRC System • 03 SRCPI • 04 DCF • 05 DPA • 06 - 99 Others
~~**deviceVerificationMethod**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	Length = 2	Verification method used to verify Consumer Device information. Valid values are: • 01 - 20 EMVCo future use • 21 - 99 SRC System specific

Data Element	R/C/O	Constraints	Description
~~**deviceVerificationResults**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	Length = 2	Indicates whether the device was verified or not, and what the results are when verified. Valid values are: • 01 Verified • 02 Not Verified • 03 Not performed • 04 - 20 EMVCo future use • 21 - 99 SRC System specific
~~**deviceVerificationTimestamp**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	UTC time in Unix epoch format	Date and time when the device verification was conducted
~~**deviceAssuranceData**~~ ~~Type: String~~ DEPRECATED	O		Data collected that is associated with the device for assurance purposes
~~**relationshipVerificationEntity**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	Length = 2	Entity performing relationship verification of a combination of data. The valid values are: • 01 SRC Initiator • 02 SRC System • 03 SRCPI • 04 DCF • 05 DPA • 06 - 99 Others
~~**relationshipVerificationMethod**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	Max Length = 2	Verification method used to verify information associated with the relationship

Data Element	R/C/O	Constraints	Description
~~**relationshipVerificationResults**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	Max Length = 2	Results of the verification of the relationship of a combination of data
~~**relationshipVerificationTimestamp**~~ ~~Type: String (Numeric)~~ DEPRECATED	O	UTC time in Unix epoch format	Date and time when the relationship verification was conducted
~~**relationshipAssuranceData**~~ ~~Type: String~~ DEPRECATED	O		Data collected that is associated with the binding relationship for assurance purposes

2.1.7 Card

Table 2.7: Card

Data Element	R/C/O	Constraints	Description
primaryAccountNumber Type: String (Numeric)	R	Min Length = 9 Max Length = 19	Primary Account Number. A variable length, ISO/IEC 7812-compliant account number that is generated within account ranges associated with a BIN by a Card Issuer
panExpirationMonth Type: String (Numeric)	C	Length = 2	Expiration month expressed as a two-digit month (MM) **Conditionality**: Required when specified for the Card (PAN)

Data Element	R/C/O	Constraints	Description
panExpirationYear Type: String (Numeric)	C	Length = 4	Expiration year expressed as a four-digit calendar year (YYYY) **Conditionality**: Required when specified for the Card (PAN)
cardSecurityCode Type: String (Numeric)	O	Length = 3 or 4	Card security code
cardholderFullName Type: String	O	Max Length = 100	Cardholder name
cardholderFirstName Type: String	O	Max Length = 50	Cardholder first name
cardholderLastName Type: String	O	Max Length = 50	Cardholder last name
billingAddress Type: Address	O	See Address	Billing address
paymentAccountReference Type: String	O	Max Length = 29	A non-financial reference assigned to each unique PAN and used to link a payment account represented by that PAN to affiliated Payment Tokens
customerServiceEmailAddress Type: String	O	Max Length = 255	Customer service email address
customerServicePhoneNumber Type: PhoneNumber	O	See PhoneNumber	Customer service phone number

Data Element	R/C/O	Constraints	Description
customerServiceUri Type: String	O	Max Length = 1024	Customer service webpage URI

2.1.8 CardholderData

Table 2.8: CardholderData

Data Element	R/C/O	Constraints	Description
fullName Type: String	O	Max Length = 100	Cardholder name
firstName Type: String	O	Max Length = 50	Cardholder first name
lastName Type: String	O	Max Length = 50	Cardholder last name
issuerIdentity Type: String	O	Max Length = 64	Cardholder identity as known by the Card Issuer. This generally enables access to an application, website or other. Examples include username/email address/mobile number
emailAddress Type: String	O	Max Length = 255	Cardholder email address. This is Cardholder generated and represents contact or notification data
mobileNumber Type: PhoneNumber	O	See PhoneNumber	Cardholder mobile phone number
billingPhoneNumber Type: PhoneNumber	O	See PhoneNumber	Cardholder billing phone number

2.1.9 CommunicationsConsent

Table 2.9: CommunicationsConsent

Data Element	R/C/O	Constraints	Description
communicationsOptIn Type: Boolean	O	Boolean	Consumer's communications opt in preference**.**
affiliateCommunicationsOptIn Type: Boolean	O	Boolean	Consumer's affiliate communications opt in preference
allowEmail Type: Boolean	O	Boolean	Consumer's preference for receiving communications via email
allowText Type: Boolean	O	Boolean	Consumer's preference for receiving communications via SMS
allowCall Type: Boolean	O	Boolean	Consumer's preference for receiving communications via voice calls
allowPush Type: Boolean	O	Boolean	Consumer's preference for receiving communications via a notification channel

2.1.10 ComplianceSettings

Table 2.10: ComplianceSettings

Data Element	R/C/O	Constraints	Description
privacy Type: Consent	O	See Consent	Consent wording for privacy policy
tnc Type: Consent	O	See Consent	Consent wording for T&Cs policy

Data Element	R/C/O	Constraints	Description
cookie Type: Consent	O	See Consent	Consent wording for cookie policy
geoLocation Type: Consent	O	See Consent	Consent wording for geolocation policy
communications Type: CommunicationsConsent	O	See Communications Consent	Indicates the Consumer's consent to receive communications

2.1.11 ~~ConfirmationData~~ DEPRECATED

Replaced by ConfirmationData2 (Section 2.1.12)

Table 2.11: ~~ConfirmationData~~ DEPRECATED

Data Element	R/C/O	Constraints	Description
~~checkoutEventType~~ ~~Type: String (Numeric)~~	O	Length = 2	Event type associated with the update. Valid values are: • 01 Authorise • 02 Capture • 03 Refund • 04 Cancel • 05 Fraud • 06 Chargeback • 07 Other

Data Element	R/C/O	Constraints	Description
~~**checkoutEventStatus**~~ ~~Type: String (Numeric)~~	O	Length = 2	Event type associated with the order. Valid values are: • 01 Created • 02 Confirmed • 03 Cancelled • 04 Fraud Cancelled • 05 Others • 06 - 50 EMVCo future use • 51 - 99 SRC System specific
~~**confirmationStatus**~~ ~~Type: String (Numeric)~~	O	Length = 2	Status of the event as provided by the SRC Initiator in the Confirmation message. Valid values are: • 01 Success • 02 Failure • 03 Other
~~**confirmationReason**~~ ~~Type: String~~	O	Max Length = 64	Description of the reason for the event associated with the order
~~**confirmationTimestamp**~~ ~~Type: String (Numeric)~~	O	UTC time in Unix epoch format	Date and time of the event completion corresponding to the Confirmation event by the SRC Initiator
~~**networkAuthorizationCode**~~ ~~Type: String~~	O	Max Length = 25	Authorisation code associated with an approved transaction
~~**networkTransactionIdentifier**~~ ~~Type: String~~	O	Max Length = 25	Unique authorisation related tracing value assigned by a Payment Network and provided in an authorisation response

Data Element	R/C/O	Constraints	Description
~~**paymentNetworkReference**~~ ~~Type: String~~	O	Max Length = 25	Transaction identifier as provided by a Payment Network after authorisation has been complete
~~**assuranceData**~~ ~~Type: AssuranceData~~	O	See AssuranceData	Assurance data
~~**transactionAmount**~~ ~~Type: TransactionAmount~~	O	See TransactionAmount	Amount of the transaction

2.1.12 ConfirmationData2

Table 2.12: ConfirmationData2

Data Element	R/C/O	Constraints	Description
checkoutEventType Type: String (Numeric)	R	Length = 2	Event type associated with the confirmation. Valid values are: • 00 Place Order • 01 Authorise • 02 Capture • 03 Refund • 04 Cancel (Auth Reversal) • 05 Fraud • 06 Chargeback • 07 Cancel before Auth • 08 Auth for account validation

Data Element	R/C/O	Constraints	Description
checkoutEventStatus Type: String (Numeric)	O	Length = 2	Event status associated with the order. Valid values are: • 01 Created • 02 Confirmed • 03 Cancelled • 04 Fraud Cancelled • 05 Others • 06 - 50 EMVCo future use • 51 - 99 SRC System specific
confirmationStatus Type: String (Numeric)	R	Length = 2	Status related to the `checkoutEventType` as provided by the SRC Initiator. Valid values are: • 01 Success • 02 Failure • 03 Other • 04 Timeout
confirmationReason Type: String	O	Max Length = 64	Description of the reason for the event associated with the order
confirmationTimestamp Type: String (Numeric)	R	UTC time in Unix epoch format	Date and time of the event set by the SRC Initiator
networkAuthorizationCode Type: String	C	Max Length = 25	Authorisation code associated with an approved transaction **Conditionality:** Required when the value of: • `checkoutEventType` is set to 01 (Authorize) or 03 (Refund); *and* • `confirmationStatus` is set to 01 (Success)

Data Element	R/C/O	Constraints	Description
networkTransactionIdentifier Type: String	O	Max Length = 25	Unique authorisation related tracing identifier assigned by a Payment Network and provided in an payment authorisation response
paymentNetworkReference Type: String	O	Max Length = 25	Transaction identifier as provided by a Payment Network payment authorisation has been completed
assuranceData Type: AssuranceData	O	See AssuranceData	Assurance data
transactionAmount Type: TransactionAmount	C	See TransactionAmount	Amount of the transaction **Conditionality:** Required when the value of: • `checkoutEventType` is set to 01 (Authorize) or 03 (Refund); *and* • `confirmationStatus` is set to 01 (Success)

2.1.13 Consent

Table 2.13: Consent

Data Element	R/C/O	Constraints	Description
acceptedVersion Type: String	O	Max Length = 10	Version accepted by the Consumer
latestVersion Type: String	O	Max Length = 10	Latest version
latestVersionUri Type: String	O	Max Length = 1024	URI of the latest version

2.1.14 Consumer

Table 2.14: Consumer

Data Element	R/C/O	Constraints	Description
consumerIdentity Type: ConsumerIdentity	R	See ConsumerIdentity	Primary verifiable Consumer Identity within an SRC Profile (e.g. an email address or a mobile phone number)
emailAddress Type: String	O	Max Length = 255	Consumer-provided email address
mobileNumber Type: PhoneNumber	O	See PhoneNumber	Consumer-provided mobile number
nationalIdentifier Type: String	O	Max Length = 20	Geographic-specific, nationally-provided identifier for the Consumer
countryCode Type: String	O	ISO 3166-1 alpha-2 country code	Consumer-provided country code
languageCode Type: String	O	ISO 639-1 Code	Consumer-provided language choice
firstName Type: String	O	Max Length = 50	Consumer-provided first name
lastName Type: String	O	Max Length = 50	Consumer-provided last name
fullName Type: String	O	Max Length = 100	Consumer-provided full name

2.1.15 ConsumerIdentity

Table 2.15: ConsumerIdentity

Data Element	R/C/O	Constraints	Description
identityProvider Type: IdentityProvider	O	See IdentityProvider	Entity or organisation that collected and verified the Consumer Identity
identityType Type: ConsumerIdentityType	R	See ConsumerIdentity Type	Type of Consumer Identity transmitted or collected
identityValue Type: String	R	Max Length = 255	Consumer Identity value that corresponds to the Consumer Identity Type

2.1.16 Dcf

Table 2.16: Dcf

Data Element	R/C/O	Constraints	Description
applicationType Type: ApplicationType	O	See ApplicationType	Type of the environment of the DCF
uri Type: String	O	Max Length = 1024	DCF URI as provided by DCF
logoUri Type: String	O	Max Length = 1024	Logo image URI provided by the DCF to support presentation
name Type: String	O	Max Length = 60	Legal Name of DCF Onboarded to the SRC System

2.1.17 DeviceData

Table 2.17: DeviceData

Data Element	R/C/O	Constraints	Description
type Type: String	O	Max Length = 255	Type of device being used. Example values are: • Mobile Phone • Tablet • Laptop • Personal Assistant • Connected Auto • Home Appliance • Wearable • Stationary Computer • E-Reader • Handheld Gaming Devices • Other
manufacturer Type: String	O	Max Length = 255	Manufacturer of the device
brand Type: String	O	Max Length = 255	Brand name of the device
model Type: String	O	Max Length = 255	Specific model of the device

2.1.18 DeliveryContactDetails

Table 2.18: DeliveryContactDetails

Data Element	R/C/O	Constraints	Description
contactFullName Type: String	O	Max Length = 100	Consumer-provided name of the contact person
contactPhoneNumber Type: PhoneNumber	O	See PhoneNumber	Consumer-provided phone number of the contact person

Data Element	R/C/O	Constraints	Description
numberIsVoiceOnly Type: Boolean	C		Indicates that the phone number provided is not capable of receiving text messages. **Conditionality:** Required when `contactPhoneNumber` is provided
contactEmailAddress Type: email	O	See Email	Consumer-provided email address of the contact person
instructions Type: String	O	Max Length = 1024	Consumer-provided delivery instructions

2.1.19 DigitalCardData

Table 2.19: DigitalCardData

Data Element	R/C/O	Constraints	Description
status Type: DigitalCardStatus	R	See DigitalCardStatus	State of the Digital Card
presentationName Type: String	O	Max Length = 64	Presentation text created by the Consumer to enable recognition of the PAN. This value is defined by the Consumer (e.g. nickname)
descriptorName Type: String	R	Max Length = 64	Presentation text defined by the SRC Programme that describes the PAN presented as a Digital Card
artUri Type: String	R	Max Length = 1024	URI that hosts the Card Art image to be used for presentation purposes. Can be provided by SRCPI

Data Element	R/C/O	Constraints	Description
artHeight Type: String (Numeric)	O		Height of the card art in pixels
artWidth Type: String (Numeric)	O		Width of the card art in pixels
pendingEvents Type: List<CardPendingEvent>	C	See CardPendingEvent	Set of events that are pending completion such as AVS or SCA **Conditionality**: Required when the value of `status` is set to PENDING

2.1.20 DigitalCardFeature

Table 2.20: DigitalCardFeature

Data Element	R/C/O	Constraints	Description
content Type: String	R	Max Length = 1024	Content of the Digital Card Feature. The value is specific for the `contentType`
contentType Type: DigitalCardFeatureContentType	R	See DigitalCardFeatureContentType	Type of the content of the Digital Card Feature
style Type: String	O	Max Length = 1024	URL of a CSS style sheet that describes how to present a Digital Card Feature
width Type: String (Numeric)	O		Width to be applied to display of a Digital Card Feature image

height Type: String (Numeric)	O		Height to be applied to display of a Digital Card Feature image

2.1.21 DigitalCardUpdateNotification

Table 2.21: DigitalCardUpdateNotification

Data Element	R/C/O	Constraints	Description
serviceId Type: String	O		Service identifier of the updated card
srcDigitalCardId Type: String	C	Max Length=36	Identifier of the updated card **Conditionality**: `srcDigitalCardId` is required if `maskedCard` is not present
authorization Type: String	O		First Party Token that may be provided if the `maskedCard` is not present
maskedCard Type: MaskedCard	C	See MaskedCard	Updated masked card data **Conditionality**: `maskedCard` is required if `srcDigitalCardId` is not present
eventTimestamp Type: String (Numeric)	R	UTC time in Unix epoch format	Date and time of the card update event
srcCorrelationId Type: String	O		SRC Correlation Id corresponding to this SRC checkout transaction. May be provided if the notification occurs during checkout
reason Type: String	O	Max length=255	Reason for the update of the card

2.1.22 DpaData

Table 2.22: DpaData

Data Element	R/C/O	Constraints	Description
dpaPresentationName Type: String	O	Max Length = 60	Merchant company name associated with the DPA to be used for presentation purposes within the user experience
dpaAddress Type: Address	O	See Address	DPA business address
dpaName Type: String	R	Max Length = 60	Legal name of Registered DPA
dpaEmailAddress Type: String	O	Max Length = 255	DPA contact email address
dpaPhoneNumber Type: PhoneNumber	O	See PhoneNumber	DPA contact phone number
dpaLogoUri Type: String	O	Max Length = 1024	URI of the logo of the DPA
dpaSupportEmailAddress Type: String	O	Max Length = 255	DPA support contact email address
dpaSupportPhoneNumber Type: PhoneNumber	O	See PhoneNumber	DPA support contact phone number
dpaSupportUri Type: String	O	Max Length =1024	DPA's support URI

Data Element	R/C/O	Constraints	Description
dpaUri Type: String	O	Max Length = 1024	A suitable unique DPA identifier. May contain the DPA business website URL or mobile application identifier in reversed domain notation or any other suitable unique DPA identifier
applicationType Type: ApplicationType	O	See ApplicationType	Type of DPA
merchantAccountInformation Type: String	O	Max Length = 1024	Implementation specific account information for an alternative acceptance channel

2.1.23 DpaTransactionOptions

Table 2.23: DpaTransactionOptions

Data Element	R/C/O	Constraints	Description
transactionAmount Type: TransactionAmount	C	See TransactionAmount	The amount of the transaction **Conditionality**: Required when 3DS is to be performed by SRC System (i.e. the value of `threeDsPreference` is set to ONBEHALF)
transactionType Type: TransactionType	O	See TransactionType	Type of transaction initiated for which the SRC System is being sent a request

Data Element	R/C/O	Constraints	Description
deliveryMethod Type: DeliveryMethod	O	See DeliveryMethod	An indication of the manner in which the purchased goods are to be delivered, independent of the `dpaBillingPreference` or `dpaShippingPreference` data elements
dpaBillingPreference Type: AddressPreference	O	See AddressPreference	Verbosity of the billing address required by the DPA
dpaAcceptedBillingCountries Type: List<String>	O	Array of country codes in ISO 3166-1 alpha-2 format	Billing restrictions. Payments from all the listed billing countries are accepted For example: ["US","CA","AU"] An empty list or the absence of this data element means that all countries are accepted.
dpaShippingPreference Type: AddressPreference	O	See AddressPreference	Verbosity of the shipping address required by the DPA

Data Element	R/C/O	Constraints	Description
dpaAcceptedShippingCountries Type: List<String>	O	Array of country codes in ISO 3166-1 alpha-2 format	Shipping restrictions. Shipping region country codes that limits the selection of eligible shipping addresses For example: ["US","CA","AU"] An empty list or the absence of this data element means that all countries are accepted.
consumerEmailAddressRequested Type: Boolean	O		Indicates whether the DPA expects the Consumer email address to be returned in the SRC Payload
consumerNameRequested Type: Boolean	O		Indicates whether the DPA expects the Consumer name to be returned in the SRC Payload
consumerPhoneNumberRequested Type: Boolean	O		Indicates whether the DPA expects the Consumer phone number to be returned in the SRC Payload
merchantCategoryCode Type: String	O	Length = 4	Describes the merchant's type of business, product or service
merchantCountryCode Type: String	O	ISO 3166-1alpha-2 country code	Country code of the merchant
merchantOrderId Type: String	O	UUID	Digital Payment Application generated order/invoice number corresponding to a Consumer purchase

Data Element	R/C/O	Constraints	Description
threeDsPreference Type: ThreeDsPreference	R	See ThreeDsPreference	Merchant's 3DS preferences
threeDsInputData Type: JSONObject	C		Merchant's 3DS input data. **Conditionality**: Required when 3DS is to be performed by SRC System (i.e. the value of `threeDsPreference` is set to ONBEHALF)
srcTokenRequestData Type: JSONObject	O		Token specific data provided by the merchant
paymentOptions Type: List<PaymentOptions>	O	See PaymentOptions	Specifies the Dynamic Data requirement for the payload creation
dpaLocale Type: String	O	ISO language country pair. [ISO 639-1 Code] [ISO 3166-1 alpha-2 country code]	Merchant's preferred locale. For example: ["en_US", "fr_CA"]
customInputData Type: JSONObject	O		Extensible container that allows DPA to pass SRC System-specific data to the SRC System
orderType Type: String	O	Length = 255	Type of the order
confirmPayment Type: Boolean	O		Default value: `false`

Data Element	R/C/O	Constraints	Description
			• DCF is expected to prompt the Consumer to confirm payment when value is set to `true` • DPA is expected to prompt the Consumer to confirm payment when value is set to `false`

2.1.24 DynamicData

Table 2.24: DynamicData

Data Element	R/C/O	Constraints	Description
dynamicDataValue Type: String	C		Value of the dynamic data **Conditionality:** Required when the value of `dynamicDataType` is not set to NONE
dynamicDataType Type: DynamicDataType	R	See DynamicDataType	Type of the Dynamic Data
dynamicDataExpiration Type: String (Numeric)	O	UTC time in Unix epoch format	Date and time at which the Dynamic Data expires

2.1.25 EnrollmentReferenceData

Table 2.25: EnrollmentReferenceData

Data Element	R/C/O	Constraints	Description
enrollmentReferenceId Type: String	R	Max Length = 256	Identifier of the enrolment reference

Data Element	R/C/O	Constraints	Description
enrollmentReferenceType Type: EnrollmentReferenceType	R	See EnrollmentReferenceType	Type of the enrolment reference
enrollmentReferenceProvider Type: String	O	Max Length = 256	Provider of the enrolment reference

2.1.26 Error

Table 2.26: Error

Data Element	R/C/O	Constraints	Description
status Type: Numeric	R	Length = 3	HTTP status code to categorise the errors
reason Type: String	R	Max Length = 32	Error reason as associated with the HTTP status code
message Type: String	R	Max Length = 255	Error message as associated with the HTTP status code
errorDetail Type: List<ErrorDetail>	O	See ErrorDetail	Error details

2.1.27 ErrorDetail

Table 2.27: ErrorDetail

Data Element	R/C/O	Constraints	Description
reason Type: String	O	Max Length = 32	Error reason
source Type: String	O	Max Length = 255	Name of the source which generated this error

Data Element	R/C/O	Constraints	Description
message Type: String	O	Max Length = 255	Error message
sourceType Type: String	O	Max Length = 32	Type of the source

2.1.28 EventHistory

Table 2.28: EventHistory

Data Element	R/C/O	Constraints	Description
ageOfSrcPanEnrolmentSinceCreated Type: String (Numeric)	O	Max Length = 5	Age, in days, of the SRC Profile as it exists in the SRC System since the time it was created
srcAgeSinceLastSuccessfulTransaction Type: String (Numeric)	O	Max Length = 5	Age, in days, of the SRC Profile as it exists in the SRC System from the time of the last successful transaction
ageOfSrcRelationship Type: String (Numeric)	O	Max Length = 5	Age, in days, of the SRC Profile in the SRC System
ageOfConsumerRelationship Type: String (Numeric)	O	Max Length = 5	Age, in days, since the Consumer profile binding event occurred at the SRC Profile

Data Element	R/C/O	Constraints	Description
billingAndShippingRelationship Type: String	O	Length = 2	Relationship between the Cardholder billing and shipping information. Valid values are: • 01 Same as Cardholder's billing address • 02 Consumer's preferred shipping address • 03 Consumer other address
shippingAddressUsageNew Type: String (Numeric)	O	UTC time in Unix epoch format	Date when the shipping address used for this transaction was first used with the SRC Initiator
ageOfShippingAddressUsage Type: String (Numeric)	O	Max Length = 5	Age, in days, since shipping address used for this transaction was first used by the SRC System

2.1.29 IdentityValidationChannel

Table 2.29: IdentityValidationChannel

Data Element	R/C/O	Constraints	Description
validationChannelId Type: String	R	Max Length = 36	Reference identifier of the validation channel
identityProvider Type: IdentityProvider	O	See IdentityProvider	Entity or organisation that can validate the identity
identityType Type: IdentityValidationChannelType	R	See IdentityValidationChannelType	Type of the identity validation channel (e.g. email, SMS)

Data Element	R/C/O	Constraints	Description
maskedValidationChannel Type: String	O	Max Length = 255	Masked identity validation channel (e.g. masked email, masked mobile number)

2.1.30 MaskedAddress

Table 2.30: MaskedAddress

Data Element	R/C/O	Constraints	Description
addressId Type: String	R	UUID	Identifier used to point to the address
name Type: String	O	Max Length = 100	Name of the individual receiving the delivered goods or service. Only applicable for the shipping address
line1 Type: String	O	Max Length = 75	Address line 1
line2 Type: String	O	Max Length = 75	Address line 2
line3 Type: String	O	Max Length = 75	Address line 3
city Type: String	O	Max Length = 50	Address city
state Type: String	O	Max Length = 30	Address state
countryCode Type: String	O	ISO 3166-1 alpha-2 country code	Address country code
zip Type: String	O	Max Length = 16	Address zip/postal code

Data Element	R/C/O	Constraints	Description
createTime Type: String (Numeric)	O	UTC time in Unix epoch format	Date and time the address was created
lastUsedTime Type: String (Numeric)	O	UTC time in Unix epoch format	Date and time the address was last used

2.1.31 MaskedCard

Table 2.31: MaskedCard

Data Element	R/C/O	Constraints	Description
srcDigitalCardId Type: String	C	Max Length=36	Reference identifier to the Digital Card representing the PAN or Payment Token **Conditionality**: • Required when returned to an SRCI or DCF • Optional when returned to an SRCPI
srcPaymentCardId Type: String	C	Max Length = 36	Reference identifier to the PAN that enables the SRC System to communicate with the SRCPI without transmitting the actual PAN. It is associated with the SRC Profile to which the Payment Card belongs and is unique within an SRC System **Conditionality**: Required when returned to the SRCPI
panBin Type: String (Numeric)	R	Max Length = PAN Length - 10	First significant digits of the PAN in an unmasked form

Data Element	R/C/O	Constraints	Description
panLastFour Type: String (Numeric)	R	Length = 4	Last four digits of the PAN in an unmasked form
tokenBinRange Type: String (Numeric)	C	Max Length = Payment Token Length - 10	Specific BIN range or subset of the BIN Range that has been designated only for the purpose of issuing Payment Tokens in an unmasked form **Conditionality**: Required when a Payment Token is used
tokenLastFour Type: String (Numeric)	C	Length = 4	Last four digits of the Payment Token in an unmasked form **Conditionality**: Required when a Payment Token is used
digitalCardData Type: DigitalCardData	R	See DigitalCardData	Contains Digital Card information that is used in the acceptance environment and user interface. It refers to the actual PAN or Payment Token without disclosing either
panExpirationMonth Type: String (Numeric)	C	Length = 2	Expiration month expressed as a two-digit month (MM) used for presentation purposes **Conditionality**: Required when specified for the card (PAN)

Data Element	R/C/O	Constraints	Description
panExpirationYear Type: String (Numeric)	C	Length = 4	Expiration year expressed as four-digit calendar year(YYYY), used for presentation purposes **Conditionality**: Required when specified for the card (PAN)
paymentCardDescriptor Type: String	O	Max Length = 32	Conveys the card brand, and will be a free-form string, to be defined within an SRC Programme
paymentCardType Type: String	O	Max Length = 32	Conveys the card type
digitalCardFeatures Type: List<DigitalCardFeature>	O	See DigitalCardFeature	Attributes related to the Digital Card Features that should be displayed to the Consumer
countryCode Type: String	O	ISO 3166-1 alpha-2 country code	Country code of issuance associated with the Card Issuer's BIN license
maskedBillingAddress Type: MaskedAddress	O	See MaskedAddress	Masked billing address associated with the card
dcf Type: Dcf	O	See Dcf	Digital Card Facilitator associated with the card
serviceId Type: String	O	Max Length = 255	Service identifier associated to an SRC System specific configuration
paymentAccountReference Type: String	O	Max Length = 29	A non-financial reference assigned to each unique PAN and used to link a payment account represented by that PAN to affiliated Payment Tokens

Data Element	R/C/O	Constraints	Description
customerServiceEmailAddress Type: String	O	Max Length = 255	Customer service email address
customerServicePhoneNumber Type: PhoneNumber	O	See PhoneNumber	Customer service phone number
customerServiceUri Type: String	O	Max Length = 1024	Customer service webpage URI
dateOfCardCreated Type: String (Numeric)	R	UTC time in Unix epoch format	Date when card was enrolled into the SRC System
dateOfCardLastUsed Type: String (Numeric)	O	UTC time in Unix epoch format	Date when card was last used for an SRC transaction

2.1.32 MaskedConsumer

Table 2.32: MaskedConsumer

Data Element	R/C/O	Constraints	Description
srcConsumerId Type: String	O	UUID	Reference identifier generated by the SRC System
maskedConsumerIdentity Type: MaskedConsumerIdentity	R	See MaskedConsumerIdentity	Masked value of the primary verifiable Consumer Identity within an SRC Profile (e.g. an email address or a mobile phone number)
maskedEmailAddress Type: String	O	Max Length = 255	Masked Consumer email address

Data Element	R/C/O	Constraints	Description
maskedMobileNumber Type: PhoneNumber	O	See PhoneNumber	Masked Consumer mobile phone number
maskedNationalIdentifier Type: String	O	Max Length = 20	Masked Consumer national identifier
complianceSettings Type: ComplianceSettings	O	See ComplianceSettings	Consumer compliance settings
countryCode Type: String	O	ISO 3166-1 alpha-2 country code	Consumer-provided country code
languageCode Type: String	O	ISO 639-1 Code	Consumer-provided language choice
status Type: ConsumerStatus	R	See ConsumerStatus	Current status of the Consumer
maskedFirstName Type: String	O	Max Length = 50	Masked Consumer first name
maskedLastName Type: String	O	Max Length = 50	Masked Consumer last name
maskedFullname Type: String	O	Max Length = 100	Masked Consumer name
dateConsumerAdded Type: String (Numeric)	R	UTC time in Unix epoch format	Date Consumer was added to the SRC System
dateConsumerLastUsed Type: String (Numeric)	O	UTC time in Unix epoch format	Date Consumer last transacted as determined by the SRC System

2.1.33 MaskedConsumerIdentity

Table 2.33: MaskedConsumerIdentity

Data Element	R/C/O	Constraints	Description
identityProvider Type: IdentityProvider	O	See IdentityProvider	Entity or organisation that collected and verifies the Consumer Identity
identityType Type: ConsumerIdentityType	R	See ConsumerIdentity Type	Type of Consumer Identity transmitted or collected
maskedIdentityValue Type: String	R	Max Length = 255	Masked Consumer Identity value (e.g. masked email address or masked mobile phone number)

2.1.34 Payload

Table 2.34: Payload

Data Element	R/C/O	Constraints	Description
card Type: Card	C	See Card	Card data associated with the PAN used for the purchase **Conditionality**: Required when the: • value of the relevant data element of type `PayloadTypeIndicator` was set to FULL or PAYMENT; *and* • SRC System determines that a PAN-based payload must be returned. A `card` is required if a `token` is not present. `card` and `token` are mutually exclusive

Data Element	R/C/O	Constraints	Description
token Type: PaymentToken	C	See PaymentToken	Payment Token data associated with the PAN used for the purchase **Conditionality**: Required when the: • Value of the relevant data element of type `PayloadTypeIndicator` was set to FULL or PAYMENT; *and* • SRC System determines that a Payment Token-based payload must be returned A `token` is required if a `card` is not present. `card` and `token` are mutually exclusive
shippingAddress Type: Address	C	See Address	Shipping address as required for the delivery of the goods/services being purchased **Conditionality**: Required when: • The value of the relevant data element of type `PayloadTypeIndicator` was set to FULL or NON_PAYMENT; *and* • Identified shipping address is available in the SRC Profile; *and* • Shipping address was requested (based on `dpaShippingPreference`)

Data Element	R/C/O	Constraints	Description
consumerEmailAddress Type: String	C	Max Length = 255	Consumer-provided email address **Conditionality**: Required when: • The value of the relevant data element of type `PayloadTypeIndicator` was set to FULL or NON_PAYMENT; *and* • Email address is available in the SRC Profile; *and* • Email address was requested (`consumerEmailAddressRequested` set to `true`)
consumerFirstName Type: String	C	Max Length = 50	Consumer-provided first name **Conditionality**: Required when: • The value of the relevant data element of type `PayloadTypeIndicator` was set to FULL or NON_PAYMENT; *and* • Consumer first name is available in the SRC Profile; *and* • Consumer name was requested (`consumerNameRequested` set to `true`)

Data Element	R/C/O	Constraints	Description
consumerLastName Type: String	C	Max Length = 50	Consumer-provided last name **Conditionality**: Required when: • The value of the relevant data element of type `PayloadTypeIndicator` was set to FULL or NON_PAYMENT; *and* • Consumer last name is available in the SRC Profile; *and* • Consumer name was requested (`consumerNameRequested` set to `true`)
consumerFullName Type: String	C	Max Length = 100	Consumer-provided name **Conditionality**: Required When: • The value of the relevant data element of type `PayloadTypeIndicator` was set to FULL or NON_PAYMENT; *and* • Consumer name is available in the SRC Profile; *and* • Consumer name was requested (`consumerNameRequested` set to `true`)

Data Element	R/C/O	Constraints	Description
consumerMobileNumber Type: PhoneNumber	C	See Phonenumber	Consumer-provided mobile number **Conditionality**: Required when: • The value of the relevant data element of type `PayloadTypeIndicator` was set to FULL or NON_PAYMENT; *and* • Consumer mobile number is available in the SRC Profile; *and* • Consumer mobile number is requested (`consumerPhoneNumberRequested` set to `true`)
srcTokenResultsData Type: JSONObject	O		SRC system specific Token data
dynamicData Type: List<DynamicData>	R	See DynamicData	Dynamic data, generated using the `dynamicDataType` preference indicated in `paymentOptions`

Data Element	R/C/O	Constraints	Description
billingAddress Type: Address	C	See Address	Billing address associated with the card used for the purchase **Conditionality**: Required when: • The value of the relevant data element of type `PayloadTypeIndicator` was set to FULL or NON_PAYMENT; *and* • Billing address is available in the SRC Profile; *and* • Billing address was requested (based on `dpaBillingPreference` or statically derived using the default configured during DPA Registration)
threeDsOutputData Type: JSONObject	C		Result of 3DS payment authentication **Conditionality:** Required when: • the value for `threeDsPreference` was set to ONBEHALF; *and* • 3DS authentication has been performed

2.1.35 PaymentOptions

Table 2.35: PaymentOptions

Data Element	R/C/O	Constraints	Description
dpaDynamicDataTtlMinutes Type: String (Numeric)	O		Requested "Time to Live" (expiry period) of the Dynamic Data, specified in minutes
dynamicDataType Type: DynamicDataType	O	See DynamicDataType	Type of Dynamic Data required in the payload

2.1.36 PaymentToken

Table 2.36: PaymentToken

Data Element	R/C/O	Constraints	Description
paymentToken Type: String	R	ISO/IEC 7812 format	Payment Token
tokenExpirationMonth Type: String (Numeric)	C	Length = 2	Expiration month expressed as a two-digit month (MM) **Conditionality**: Required when specified for the Payment Token
tokenExpirationYear Type: String (Numeric)	C	Length = 4	Expiration year expressed as a four-digit calendar year (YYYY) **Conditionality**: Required when specified for the Payment Token
cardholderFullName Type: String	O	Max Length = 100	Cardholder name

Data Element	R/C/O	Constraints	Description
cardholderFirstName Type: String	O	Max Length = 50	Cardholder first name
cardholderLastName Type: String	O	Max Length = 50	Cardholder last name
paymentAccountReference Type: String	O	Max Length = 29	A non-financial reference assigned to each unique PAN and used to link a payment account represented by that PAN to affiliated Payment Tokens

2.1.37 PhoneNumber

Table 2.37: PhoneNumber

Data Element	R/C/O	Constraints	Description
countryCode Type: String	R	Min Length = 1 Max Length = 4	Phone number country code as defined by the International Telecommunication Union
phoneNumber Type: String	R	Min Length = 4 Max Length = 14	Phone number without country code

2.1.38 SrcProfile

Table 2.38: SrcProfile

Data Element	R/C/O	Constraints	Description
maskedCards Type: List<MaskedCard>	O	See MaskedCard	Masked card data associated with the SRC Profile

Data Element	R/C/O	Constraints	Description
maskedShippingAddresses Type: List<MaskedAddress>	O	See MaskedAddress	Masked shipping address data associated with the SRC Profile
maskedConsumer Type: MaskedConsumer	C	See MaskedConsumer	Masked Consumer data associated with the SRC Profile **Conditionality**: Required for non-device bound SRC Profiles
authorization Type: String	R		First party authorisation token as defined in Section 5.1.4 Authorisation

2.1.39 TransactionAmount

Table 2.39: TransactionAmount

Data Element	R/C/O	Constraints	Description
transactionAmount Type: Number	R	Max Length = 18	Amount of the transaction represented as a floating-point number
transactionCurrencyCode Type: String	R	ISO 4217 currency code	Currency in which the transaction amount is expressed. It is up to the SRC Programme to determine whether the currency code is: • Alphabetic • Numeric • Both
additionalAmounts Type: List<AdditionalAmount>	O		A list of additional amounts related to the transaction

2.1.40 VerificationData

Table 2.40: VerificationData

Data Element	R/C/O	Constraints	Description
verificationType Type: VerificationType	R	See VerificationType	Type of the verification data
verificationEntity Type: String (Numeric)	R	Length = 2	Entity performing the verification See Table 2.41
verificationEvents Type: List<String (Numeric)>	O	Array of two digit codes as defined in Table 2.41	Event where the verification occurred See Table 2.41
verificationMethod Type: String (Numeric)	R	Length = 2	Method of the verification See Table 2.41
verificationResults Type: String (Numeric)	R	Length = 2	Result of the verification See Table 2.41
verificationTimestamp Type: String (Numeric)	R	UTC time in Unix epoch format	Date and time when the verification was conducted
additionalData Type: String	O		Data collected during the verification process

The `VerificationData` structure can contain data relating to various entities within the SRC Specifications. Table 2.41 provides valid values for individual attributes of the structure, depending on the type of the verification.

Table 2.41: VerificationData Values

Verification Type	Verification Entity	Verification Event	Verification Method	Verification Results
CARD	Entity performing card verification. Valid values are: • 01 SRC Initiator • 02 SRC System • 03 SRCPI • 04 DCF • 05 DPA • 06 – 99 Others	Event where the verification occurred. Valid values are: • 01 – 20 EMVCo future use • 21 – 99 SRC System specific	Validates that the PAN is active and valid at the Card Issuer. Valid values are: • 01 $0 authorisation, or single unit of currency authorisation • 02 Card Verification Number validation • 03 Postal code and address verification, where supported • 04 - 09 EMVCo future use • 10 Card Issuer Account Verification • 11 Card Issuer Interactive Cardholder Authentication – 1 Factor • 12 Card Issuer Interactive Cardholder Authentication – 2 Factor • 13 Card Issuer Risk Oriented Non-Interactive Cardholder Authentication • 14 Card Issuer Asserted Authentication • 15 – 20 EMVCo future use • 21 – 99 SRC System specific	Verification status of the PAN. Valid values are: • 01 Verified • 02 Not Verified • 03 Not performed • 04 Not Required • 05 – 20 EMVCo future use • 21 – 99 SRC System specific

Verification Type	Verification Entity	Verification Event	Verification Method	Verification Results
CARDHOLDER	Entity performing Cardholder authentication. Valid values are: • 01 SRC Initiator • 02 SRC System • 03 SRCPI • 04 DCF • 05 DPA • 06 – 99 Others	Event where the verification occurred. Valid values are: • 01 Payment transaction • 02 Add card/Card enrolment • 03 SRC Profile Access • 04 Account Verification • 05 – 20 EMVCo future use • 21 – 99 SRC System specific	Card Issuer verification of the Cardholder. Valid values are: • 01 Use of a 3-D Secure ACS • 02 Mobile banking verification of the Cardholder with an authentication code • 03 Federated login systems • 04 A shared secret between the Card Issuer and the Cardholder such as One Time Passcode (OTP), activation code • 05 No authentication • 06 Proprietary method of authentication • 07 FIDO2 • 08 – 20 EMVCo future use • 21 – 99 SRC System specific	Indicates whether the Cardholder was verified or not, and what the results are when verified. Valid values are: • 01 Verified • 02 Not Verified • 03 Not performed • 04 Not required • 05 – 20 EMVCo future use • 21 – 99 SRC System specific

Verification Type	Verification Entity	Verification Event	Verification Method	Verification Results
CONSUMER	Entity performing Consumer verification. Valid values are: • 01 SRC Initiator • 02 SRC System • 03 SRCPI • 04 DCF • 05 DPA • 06 – 99 Others	Event where the verification occurred. Valid values are: • 01 – 20 EMVCo future use • 21 – 99 SRC System specific	Verification method used to verify the Consumer credential. Valid values are: • 01 Static Passcode • 02 SMS One Time Passcode (OTP) • 03 Keyfob or EMV cardreader One Time Passcode (OTP) • 04 Application One Time Passcode (OTP) • 05 One Time Passcode (OTP) Other • 06 Knowledge Based Authentication (KBA) • 07 Out of Band Biometrics • 08 Out of Band Login • 09 Out of Band Other • 10 Risk-Based • 11 Other • 12 – 99 EMVCo future use	Indicates whether the Consumer was verified or not, and what the results are when verified. Valid values are: • 01 Verified • 02 Not Verified • 03 Not performed • 04 Not Required • 05 – 20 EMVCo future use • 21 – 99 SRC System specific

Verification Type	Verification Entity	Verification Event	Verification Method	Verification Results
DEVICE	Entity performing Device verification. Valid values are: • 01 SRC Initiator • 02 SRC System • 03 SRCPI • 04 DCF • 05 DPA • 06 – 99 Others	Event where the verification occurred. Valid values are: • 01 – 20 EMVCo future use • 21 – 99 SRC System specific	Verification method used to verify Consumer Device information. Valid values are: • 01 App Binding (App Instance ID) • 02 – 20 EMVCo future use • 21 – 99 SRC System specific	Indicates whether the device was verified or not, and what the results are when verified. Valid values are: • 01 Verified • 02 Not Verified • 03 Not performed • 04 Not Required • 05 – 20 EMVCo future use • 21 – 99 SRC System specific
RELATIONSHIP	Entity performing relationship verification of a combination of data. Valid values are: • 01 SRC Initiator • 02 SRC System • 03 SRCPI • 04 DCF • 05 DPA • 06 – 99 Others	Event where the verification occurred. Valid values are: • 01 – 20 EMVCo future use • 21 – 99 SRC System specific	Verification method used to verify information associated with the relationship.	Results of the verification of the relationship of a combination of data.

2.2 Individual Attributes

Table 2.42: Individual Attributes

Name	Constraints	Description
appInstanceId Type: String	Max Length=255	Long-lived First Party Token representing an app bound to the SRC Profile
cardDeletionReason Type: CardDeletionReason	See CardDeletionReason	Indicates the reason the card is being deleted
cardSource Type: Origin	See Origin	Indicates the entity performing the Enrolment
checkoutRequestUri Type: String	Max Length = 1024	Redirection URI for the SRCI
checkoutResponseUri Type: String	Max Length = 1024	Redirection URI for the DCF
consumerPresent Type: Boolean		Indicates if the Identity Lookup operation was successful or not
customInputData Type: JSONObject		SRC System--specific input data
customOutputData Type: JSONObject		SRC System--specific output data
dcfActionCode Type: DcfActionCode	See DcfActionCode	DCF action code
eventTimeStamp Type: String (Numeric)	UTC time in Unix epoch format	When card update event occurred

Name	Constraints	Description
idLookupSessionId Type: String	UUID	Session identifier returned by SRC System following an Identity Lookup operation. Can be used in subsequent Initiate Identity Validation operation
idValidationSessionId Type: String	UUID	Session identifier returned by SRC System following an Initiate Identity Validation operation. Used in subsequent Complete Identity Validation operation
maskedValidationChannel Type: String	Max Length = 1024	Masked value of the channel (e.g. email / phone) that the SRC System used to deliver the validation data
payloadTypeIndicatorCheckout Type: PayloadTypeIndicator	See PayloadTypeIndicator	Type of encrypted payload to be returned in the Checkout operation response
payloadTypeIndicatorPayload Type: PayloadTypeIndicator	See PayloadTypeIndicator	Type of encrypted payload to be created for the retrieval by the Get Payload operation
reason Type: String	Max Length = 255	Reason that the card update is occurring
recipientId Type: String	Max Length = 36	Recipient of the encrypted payload known to the SRC System (as provided in the Checkout operation response) for the intended recipient

Name	Constraints	Description
recipientIdCheckout Type: String	Max Length = 36	Recipient of the encrypted payload known to the SRC System (as provided in the Checkout operation response) for the intended recipient
recipientIdPayload Type: String	Max Length = 36	Recipient of the encrypted payload known to the SRC System (as retrieved by the Get Payload operation) for the intended recipient
recognized Type: Boolean		Flag indicating whether the Consumer Device (e.g. browser or client application) is recognised by the SRC System
requestedValidationChannelId Type: String	Max Length = 256	Identifier of the channel over which the identity validation should be initiated
requestor Type: Origin	See Origin	Indicates the entity requesting deletion of Digital the Card
setAsShippingAddress Type: Boolean		If set to `true`, the shipping address is also created and is set to the same as the billing address
srcClientId Type: String	Max Length=255	Reference identifier of the connecting client
srcCorrelationId Type: String	Max Length = 256	Unique identifier generated by an SRC System
srcDpaId Type: String	Max Length=255	Reference identifier of the DPA

Name	Constraints	Description
scriActionCode Type: SrciActionCode	See SrciActionCode	SRC Initiator action code
srcInitiatorId	Max Length=255	Reference identifier of the SRCI
srciTransactionId Type: String	Max Length=255	Transactional identifier provided by the SRCI
shippingAddressId Type: String	Max Length = 256	Shipping address reference identifier
threeDsInputData Type: JSONObject		Input data for 3DS processing
threeDsOutputData Type: JSONObject		Output data following 3DS processing
unbindAppInstance Type: Boolean		Flag indicating whether the Consumer has chosen to be ‘un-remembered’ from the Consumer Device
validationData Type: String	Max Length = 255	Validation data (e.g. OTP) as entered by the Consumer as a part of the step up authentication
validationMessage Type: String	Max Length = 255	Validation message that needs to be presented to the Consumer for step up authentication
version Type: String		SRC system specific versioning
windowRef Type: Window		A handle provided by the user agent to facilitate the DCF opening a custom URI in the popup/iframe window

2.3 Enumerations

Note: the enumeration values set out below are not exhaustive. Other values may be added in future versions of this SRC API Specification, or may be defined within the scope of a specific implementation.

Table 2.43: Enumerations

Name	Valid Values
AcceptanceChannelType	• EMV_MERCHANT_PRESENTED_MODE
AcceptanceChannelTechnology	• QR_CODE
AdditionalAmountType	• TIP • CONVENIENCE_FEE • SUB_TOTAL
AddressPreference	• NONE • FULL • POSTAL_COUNTRY
ApplicationType	• WEB_BROWSER • MOBILE_APP • IOT_DEVICE • OTHER
CardDeletionReason	• SUSPECTED_FRAUD • ACCOUNT_CLOSED
CardEventType	• MODIFIED • DELETED • SUSPEND • UNSUSPEND
CardPendingEvent	• PENDING_AVS • PENDING_SCA • PENDING_CONSUMER_IDV
ConsumerIdentityType	• EMAIL_ADDRESS • MOBILE_PHONE_NUMBER

Name	Valid Values
ConsumerStatus	• ACTIVE • SUSPENDED • LOCKED
DcfActionCode	• COMPLETE • CHANGE_CARD • ADD_CARD • SWITCH_CONSUMER • CANCEL • ERROR
DeliveryMethod	• NO_DELIVERY • ADDRESS_BILLING • ADDRESS_ON_FILE • ADDRESS_OTHER • PICKUP • ELECTRONIC
DigitalCardFeatureContentType	• TEXT_STRING • IMAGE_URL • CONTENT_URL • LINK_URL
DigitalCardStatus	• ACTIVE • SUSPENDED • EXPIRED • PENDING • CANCELLED
DynamicDataType	• CARD_APPLICATION_CRYPTOGRAM_SHORT_FORM • CARD_APPLICATION_CRYPTOGRAM_LONG_FORM • DYNAMIC_CARD_SECURITY_CODE • CARDHOLDER_AUTHENTICATION_CRYPTOGRAM • NONE
EnrollmentReferenceType	• SRC_DIGITAL_CARD_ID • SRC_PAYMENT_CARD_ID
IdentityProvider	• SRC

Name	Valid Values
IdentityValidationChannelType	• EMAIL • SMS • OUT_OF_BAND
Origin	• CARDHOLDER • MERCHANT • ISSUER
PayloadTypeIndicator	• SUMMARY • FULL • PAYMENT • NON_PAYMENT
SrciActionCode	• NEW_USER • AUTH_FAILED • AUTH_SKIPPED
ThreeDsPreference	• NONE • SELF • ONBEHALF
TransactionType	• PURCHASE • BILL_PAYMENT • MONEY_TRANSFER • DISBURSEMENT • P2P
VerificationType	• CARD • CARDHOLDER • CONSUMER • DEVICE • RELATIONSHIP

2.4 Signed Checkout Objects

2.4.1 Checkout Request JWS

The Checkout Request JWS is a signed object with protection of a nonce (jti) and expiry (exp) generated by the SRC System for the SRCI front-end to pass to the DCF front-end. The SRC

System can subsequently recognise/verify this JWS when it is provided by the DCF front-end in the Checkout operation.

Table 2.44: Checkout Request JOSE Header

Parameter Name	R/C/O	Description
alg	R	Algorithm used to digitally sign the payload according to RFC 7518 Section 3.1: • 'None' is not supported. • 'PS256' is preferred to 'RS256' following the recommendation in RFC 3447
kid	R	Key ID of the cryptographic public key of the signing SRC System. Relying party SHOULD use the key ID to select the appropriate key to verify the signature. The key type of the public key identified by the key ID MUST match the type of the signing algorithm.

Table 2.45: Checkout Request Claim Set

Claim Name	Cardinality	Notes
iss	1	Value has to be URI or other identifier of the SRC System that generated this JWS. The format of the identifier is specific to SRC Programme. Sample value of the URI: https://srcsystem1.com
exp	1	Expiration time in UTC and unix/epoch format. This is useful for the cases where the transaction is abandoned and the JWS can be used for one-time attack, where jti cannot help.
iat	1	Issuance time in UTC and unix/epoch format Time at which the JWS was issued. This should not be before the current date/time.

Claim Name	Cardinality	Notes
jti	0..1	Provides a unique identifier for the JWS. The value is a case-sensitive string. This helps against replay attacks
jti_IDToken	0..1	Populated from the `idToken`_JWT.jti, if the authorisation is the `idToken`
srcInitiatorId	1	Identifier of the SRCI assigned during Onboarding Type: String
maskedCard	1	Masked Digital Card information Type: `MaskedCard`
maskedConsumer	0..1	Masked Consumer information Type: `MaskedConsumer`
maskedShippingAddresses	0..n	Array of masked shipping addresses Type: List<`MaskedAddress`>
authorization	0..1	First Party Token Type: String
srcCorrelationId	1	Unique identifier corresponding to the present checkout session. A new one is generated by the SRC System if not provided in the input Type: String
srciTransactionId	0..1	Transaction-unique identifier assigned by the SRCI. Populated if provided in the input Type: String
srcDpaId	0..1	Identifier of the DPA. Populated if provided in the input Type: String
dpaData	0..1	Data associated with the DPA Type: `DpaData`
dpaTransactionOptions	1	Transaction options as provided by the DPA Type: `DpaTransactionOptions`

Claim Name	Cardinality	Notes
assuranceData	0..1	Assurance data related to the checkout flow. Populated if provided in the input Type: `AssuranceData`
checkoutRequestUri	1	The URI that the SRCI will use to invoke the DCF. This can be same as or derived from the `checkoutRequestUri` in the request body Type: String
checkoutResponseUri	1	The URI that the DCF will use to redirect back to the SRCI after the transaction is completed or cancelled or failed. Provided by SRCI during Onboarding Type: String
serviceId	0..1	Service identifier Type: String
payloadTypeIndicatorCheckout	0..1	Type of encrypted payload to be returned in the Checkout operation response Type: `PayloadTypeIndicator`
payloadTypeIndicatorPayload	0..1	Type of encrypted payload to be created for the retrieval by the Get Payload operation Type: `PayloadTypeIndicator`
recipientIdCheckout	0..1	Recipient identifier of the encrypted payload known to the SRC System (as provided in the Checkout operation response) for the intended recipient Type: String
recipientIdPayload	0..1	Recipient identifier of the encrypted payload known to the SRC System (as retrieved by the Get Payload operation) for the intended recipient Type: String

2.4.2 Checkout Payload Response

Table 2.46 defines a data type of `CheckoutPayloadResponse`.

Table 2.46: Checkout Payload Response

Data Element	R/C/O	Constraints	Description
srcCorrelationId Type: String	C		Unique identifier corresponding to the present checkout session **Conditionality**: • Required for the Checkout and Make Payment operations • Optional for the Get Payload operation
srciTransactionId Type: String	C		Transaction-unique identifier assigned by the SRCI **Conditionality**: Required when received in the request
srcDpaId Type: String	O		Identifier of the DPA generated by SRC System based on the previously provided `dpaData` or generated during the DPA Registration process
dpaData Type: DpaData	C	See DpaData	Data associated with the DPA **Conditionality**: • Required for the Make Payment operation: • Optional for the Checkout and Get Payload operations

Data Element	R/C/O	Constraints	Description
maskedConsumer Type: MaskedConsumer	C	See MaskedConsumer	Masked Consumer data associated with the SRC Profile **Conditionality**: • Required for the Checkout and Get Payload operations if the associated SRC Profile contains Consumer data • Optional for the Make Payment operation
maskedCard Type: MaskedCard	C	See MaskedCard	Masked card data **Conditionality**: • Required for the Checkout and Get Payload operations • Optional for the Make Payment operation
shippingAddressZip Type: String	C	Max Length = 16	Zip or postal code of selected shipping address **Conditionality**: Required, depending on the `dpaShippingPreference` option in the `dpaTransactionOptions` structure and if either a `shippingAddressId` or `shippingAddress` object was present in the Checkout operation request

Data Element	R/C/O	Constraints	Description
shippingAddressCountryCode Type: String	C	ISO 3166-1 alpha-2 country code	Country code of selected shipping address **Conditionality**: Required, depending on the `dpaShippingPreference` option in the `dpaTransactionOptions` structure and if either a `shippingAddressId` or `shippingAddress` object was present in the Checkout operation request
customOutputData Type: JSONObject	O		SRC System-specific data
assuranceData Type: AssuranceData	O	See AssuranceData	Assurance data related to the checkout flow
eventHistory Type: EventHistory	O	See EventHistory	Event history related to the checkout flow
~~**payload**~~ ~~Type: JWE<JWS<Payload>>~~ DEPRECATED	C	See Payload	SRC Payload. Signed by prior to being encrypted for the specific recipient **Conditionality**: Refer to the response definitions for the Checkout operation (see Section 5.5.2 Checkout) and the Get Payload operation (see Section 5.5.3 Get Payload)

Data Element	R/C/O	Constraints	Description
encryptedSignedPayload Type: JWE<JWS<Payload>>	C	See Payload	SRC Payload. Signed by prior to being encrypted for the specific recipient **Conditionality**: Refer to the response definitions for the Checkout operation (see Section 5.5.2 Checkout) and the Get Payload operation (see Section 5.5.3 Get Payload) `encryptedSignedPayload` and `encryptedPayload` are mutually exclusive
encryptedPayload Type: JWE<Payload>	C	See Payload	SRC Payload. Encrypted for the specific recipient **Conditionality**: Refer to the response definitions for the Checkout operation (see Section 5.5.2 Checkout) and the Get Payload operation (see Section 5.5.3 Get Payload) `encryptedSignedPayload` and `encryptedPayload` are mutually exclusive
dpaTransactionOptions Type: DpaTransactionOption	O	See DpaTransactionOptions	Transaction options as provided by the DPA
acceptanceChannelRelatedData Type: AcceptanceChannelRelatedData	O	See AcceptanceChannelRelatedData	Data related to the acceptance channel

2.4.3 JWS JOSE Header

The JWS structure for the signed data elements of type `CheckoutPayloadResponse` and `Payload` contains the protected JOSE header as specified in Table 2.47.

Table 2.47: JWS JOSE Header

Parameter Name	R/C/O	Description
alg	R	Algorithm used to digitally sign the payload according to RFC 7518 Section 3.1: • 'None' is not supported. • 'PS256' is preferred to 'RS256' following the recommendation in RFC 3447
kid	R	Key ID of the cryptographic public key of the signing SRC System. Relying party SHOULD use the key ID to select the appropriate key to verify the signature. The key type of the public key identified by the key ID MUST match the type of the signing algorithm
iss	R	Issuer identifier. The value is a case sensitive URI using the https scheme that contains scheme and full qualified domain name of the host only. Sample value of the URI: https://srcsystem1.com
jti	R	A pseudo-random value used as nonce. The value is a case-sensitive string
iat	R	Issuance timestamp in UTC and Unix/epoch format

2.5 Masking Rule

All masked objects should follow the masking rules as defined in the SRC Core Specification.

3 Federated Identity

The concept of federated identity enables collaborating SRC Systems to reduce friction by sharing the results of a successfully validated Consumer Identity. This Section describes a federated token that supports the notion of federated digital identity and authorisation. A Federated ID Token is issued by the SRC System as a digitally signed attestation that the identity of the requestor has been validated.

3.1 Authorisation Token

By default, the digital authorisation is a JSON Web Token (JWT) in line with RFC 7519 and compatible with OpenID Connect ID Token.

Each token needs to be digitally signed by the SRC System that issued the token. Relying parties (e.g. other SRC Systems) need to be able to validate this token using the issuing SRC System's public key. Signature has to be compliant with JSON Web Signature (JWS) specification RFC 7515.

3.1.1 Token Header

The header of the JWT has to be compliant with the JOSE Header as specified by RFC 7519. Table 3.1 describes the JOSE Header, in accordance with RFC.

Table 3.1: JOSE Header

Parameter Name	R/C/O	Description
alg	R	Algorithm used to digitally sign the payload according to RFC 7518 Section 3.1: • 'None' is not supported • 'PS256' is preferred to 'RS256' following the recommendation in RFC 3447
kid	R	Key ID for the SRC System public key to be used to verify the signature. Relying party SHOULD use the Key ID to select the appropriate key to verify the signature. The key type of the public key identified by the Key ID MUST match the type of the signing algorithm

Parameter Name	R/C/O	Description
typ	R	Media type of the token. For JWT tokens the value should be `JWT+ext.id_token`

3.1.2 Token Claims

The Federated ID Token represents digitally signed attestation that a Consumer has been identified by an SRC System. The token contains Consumer Identities that allow other SRC Systems to identify the corresponding SRC Profile.

Table 3.2: Federated ID Token Claim Set

Claim Name	Cardinality	Notes
Public Claims		
iss	1	Issuer identifier for the Issuer of the response. Identifiers MUST BE in the form of case sensitive URI using the https scheme that contains scheme and full qualified domain name of the host only. Sample value of the URI: `https://srcsystem1.com`
sub	1	Subject Identifier. A locally unique and never reassigned identifier within the Issuer for the end user (Consumer), which is intended to be consumed by the Client, e.g., `24400320` or `AItOawyewNvutrJUqsvl6qs7A4`. It MUST NOT exceed 255 ASCII characters in length. The sub value is a case sensitive string. SRC System-specific primary identifier of the Consumer that MAY BE used to locate Consumer's SRC Profile.
aud	1..n	JSON Array of the audience(s) that this ID Token is intended for.

Claim Name	Cardinality	Notes
		It MUST contain the identifier of the requestor (SRCI or DCF) as the first element of the array. It MUST also contain identifiers for participating SRC Systems as audiences. Identifiers MUST BE in the form of case sensitive URIs using the https scheme that contains scheme and full qualified domain name of the host only. Sample value of the array: ["https://srci.com", ”https://srcsystem1.com”, “https:// srcsystem2.com”, “https:// srcsystem3.com”]
exp	1	Expiration time on or after which the ID Token SHOULD NOT be accepted for processing. The processing of this parameter requires that the current date/time MUST be before the expiration date/time listed in the value. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time. See RFC 3339 [RFC3339] for details regarding date/times in general and UTC in particular. Minimum expiration timestamp SHOULD BE 15 minutes from the issued-at timestamp.
iat	1	Time at which the ID Token was issued. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time. This MAY BE before current date/time, i.e., an SRC System may cache tokens up to close to the expiration time of the token.
jti	0..1	The "jti" (JWT ID) claim provides a unique identifier for the JWT. The value is a case-sensitive string.

Claim Name	Cardinality	Notes
auth_time	0..1	Time when the end user authentication occurred. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time. Value of the claim reflects the time when the user actually provided the credentials for authentication on this specific browser or app instance: A validated token MUST be issued if and only if a channel was validated on this specific browser or app instance.
amr	0..n	List of methods end user was authenticated with. JSON array of strings that are identifiers for authentication methods used in the authentication. For instance, values might indicate that both password and OTP authentication methods were used. The amr value is an array of case sensitive strings. The authentication method used when the user actually provided the credentials for authentication on this specific browser or app instance: A validated token MUST be issued if and only if a channel was validated on this specific browser or app instance. For the specific details of each of the values, see: https://tools.ietf.org/html/draft-ietf-oauth-amr-values-04#page-3 Additional values for SRC are: • `sms_otp` • `email_otp`

Claim Name	Cardinality	Notes
Standard ID Token Claims		
phone_number	0..1	Obfuscated end user's preferred mobile phone number. Underlying phone number value MUST conform with E.164 [E.164] format except that the leading "+" special character MUST be excluded. Used by Relying Party to help to identify a matching SRC Profile. The Relying Party MUST NOT rely upon this value being unique.
phone_number_verified	0..1	`true` if the end user's phone number has been verified; otherwise `false`. When this claim value is `true`, this means that the OP (OpenID Provider) took affirmative steps to ensure that this phone number was controlled by the end user at the time the verification was performed. The means by which a phone number is verified is context-specific, and dependent upon the trust framework or contractual agreements within which the parties are operating. The Relying Party MUST NOT rely upon this value being unique. For SRC, this value MUST be `true` only if the OP can deterministically confirm that the phone number was verified by the user authenticated on this specific browser or app instance.
email	0..1	Obfuscated end user's preferred e-mail address. Underlying email address value MUST conform to the RFC 5322 addr-spec syntax simplified to all lowercase characters. Used by Relying Party to help to identify a matching SRC Profile. The Relying Party MUST NOT rely upon this value being unique.

Claim Name	Cardinality	Notes
email_verified	0..1	`true` if the end user's e-mail address has been verified; otherwise `false`. When this claim value is `true`, this means that the OP took affirmative steps to ensure that this e-mail address was controlled by the end user at the time the verification was performed. The means by which an e-mail address is verified is context-specific, and dependent upon the trust framework or contractual agreements within which the parties are operating. For SRC, this value MUST be `true` only if the OP can deterministically confirm that the email address was verified by the user authenticated on this specific browser or app instance.
Private Claims		
src_phone_number_mask	0..1	Masked Consumer mobile phone number. This MUST use E.164 format with SRC-specific masking rules. Used by the Relying Party to properly render the UI and allow a frictionless user experience.
src_email_mask	0..1	Masked Consumer e-mail address in RFC 5322 format with SRC-specific masking rules. Used by the Relying Party to properly render the UI and allow a frictionless user experience.

3.1.3 Notes on Authentication

Note that:

- The `auth_time` claim is not correlated with the `iat` claim
- The `amr` claim is optional. If value is not specified, the end user cannot be assumed as authenticated
- The `auth_time` claim SHOULD BE present only if the `amr` claim is present
- The `auth_time` claim MUST always represent the time at which a consumer-interactive authentication method was performed (e.g. `email_otp` or `sms_otp`)

- The `auth_time` claim MUST NOT represent a consumer-transparent authentication method (e.g. swk or rbd)
- The `amr` claim array MUST present authentication methods in order from oldest to most recent
- When the `amr` claim contains a list of different authentication methods the `auth_time` claim shall correspond to the most recent interactive authentication method from the `amr` list

4 SRCI – DCF Interaction

As part of a checkout flow, an SRCI may be required to invoke the DCF to support necessary aspects of the checkout user experience. Upon completion of these steps, the DCF can return control back to the SRCI.

The SRCI and DCF will not know each other directly, however. The URIs will be provided by the SRC System to the SRCI and the DCF to invoke each other for native or browser environments.

The following are the use cases to be addressed here:

- The recognized Consumer (with an `idToken`), card list is presented by the SRCI and the Consumer chooses a card. The appropriate DCF for that Digital Card is invoked by the SRCI using the URI provided by the SRC System
- The recognized Consumer (with an `idToken`), card list is presented by the SRCI and the Consumer chooses to add a new card. The appropriate default DCF for that SRC System is invoked by the SRCI using the URI provided by the SRC System
- The unrecognized Consumer (no `idToken`) adds a new card. The appropriate default DCF for that SRC System is invoked by the SRCI using the URI provided by the SRC System

All the above use cases should be addressed for the following:

- Browser and native (iOS/Android) use cases
- Support the following action/result scenarios from DCF to SRCI:
  - Change Consumer
  - Change Card
  - Add Card
  - Cancel Checkout
  - Successful Checkout
  - Error

4.1 Interaction Mechanisms

There can be various possible technical implementation approaches to support these interactions. The example flow is illustrative only to help the reader understand the concept, however actual implementations will differ due to security principles and policies.

The sequence of calls are as follows:

- SRCI front end (e.g. JavaScript from the SRCI that executes in the Consumer's browser) calls the SRC System back end to create the Checkout Request JWS and get the DCF URI
- The SRCI front end launches the DCF front end (e.g. JavaScript from the DCF that executes in the Consumer's browser) using the `checkoutRequestUri`
- After the transaction is completed, the DCF front end sends control to the SRCI using the `checkoutResponseUri` obtained from the Checkout Request JWS

4.2 Launch The DCF

This is the mechanism for the SRCI to launch a DCF from the given DCF URI using the Checkout Request JWS.

{checkoutRequestUri}?action={actionCode}&IDToken={idToken}#{checkoutRequestJws}

The `actionCode`, if passed from the SRCI to the DCF, is expected to be one of the following:

- NEW_USER: if specified, will advise the DCF that the Consumer entered the flow for Enrolment
- AUTH_FAILED: if specified, will advise the DCF that the Consumer failed identity validation with no attempts remaining
- AUTH_SKIPPED: if specified, will advise the DCF that the Consumer chose to skip identity validation

The DCF application would need to read the Checkout Request JWS from the URI fragment using document.location (in case of the browser) or using native code (in case of a native mobile app). Note that the `idToken` might not be present in scenarios such as an unrecognised Consumer adding a new card.

The usage of fragment has the benefit of not having to pass the contents of Checkout Request JWS through the network.

There can be more suitable methods like Android Intents to launch the DCF for certain native environments like Android. In those cases, the implementer can choose to use those platform-specific methods.

4.3 Redirect back to SRCI

After the transaction is processed, the control needs to be handed back to the SRCI from the DCF.

This is done using the `checkoutResponseUri` derived from the above-mentioned JWS.

For non-error scenarios:

`{checkoutRequestJws.checkoutResponseUri}?action={actionCode}&IDToken={idToken}#{checkoutResponse}`

The `checkoutResponse` is the signed data element of type `CheckoutPayloadReponse` as returned by the Checkout operation. Note that `checkoutResponse` will be present only when the content of `actionCode` has a value of COMPLETE.

The `idToken` is conditional in the response URI fragment and is present only when the Consumer successfully completes identity validation and the Consumer chooses to add/change card.

There might be more suitable methods like Android Intents to redirect back to the SRCI for certain native environments like Android. In those cases, the implementer can choose to use those platform-specific methods.

The valid values of `actionCode` are as follows:

- COMPLETE: DCF processing completed normally
- CHANGE_CARD: Consumer wishes to select an alternative card
- ADD_CARD: Consumer wishes to add a new card
- SWITCH_CONSUMER: Consumer wishes to change account profile / identity
- CANCEL: Consumer wishes to cancel the flow
- ERROR: an error was detected and the DCF processing cannot continue

For error scenarios:

`{checkoutRequestJws.checkoutResponseUri}?action=ERROR&error={errorCode}&errorDescription={errorDescription}`

The error codes and description values are defined in Table 4.1.

Table 4.1: Error Codes

<table>
<tr><th>Name</th><th>R/C/O</th><th>Description</th></tr>
<tr><td>errorCode
Type: string</td><td>R</td><td>Code for the error. Used by the API client for error handling.
<table>
<tr><th>error</th><th>Comments</th></tr>
<tr><td>TERMS_AND_CONDITIONS_NOT_ACCEPTED</td><td>Terms and Conditions are not accepted</td></tr>
</table>
</td></tr>
</table>

		ACCT_INACCESSIBLE	User account is disabled or locked out
		AUTH_INVALID	Client is not authorised to make this request
		AUTH_ERROR	Unrecognised client
		SERVICE_ERROR	Unexpected server error
		INVALID_REQUEST	This error can result when the `checkoutRequestJws` format or contents are invalid (due to invalid signature, etc.)
errorDescription Type: string	O	Description of the error message. Should not be used for display purposes since this message is not localised. However, it could be used for logging and debugging purposes.	

5 Server-Side API

5.1 API Principles

- The server-side API is designed as a set of web services where each API endpoint represents an operation to be performed
- All request and response data elements are sent in the JSON (JavaScript Object Notation) data-interchange format
- Each endpoint in the API specifies the HTTP Method used to perform the required operation
- All data elements or parameters of type String in requests and responses, or within complex data objects are UTF-8 encoded
- All actionable fields MUST be provided as part of the request parameters (path, query or body). Only meta data must be carried in the headers. This ensures that the SDK and API spec have similar function signatures and that actionable fields can be included as part of cryptographic signatures to control against data tampering as well as repudiation claims

5.1.1 Common HTTP Status Codes

The following common HTTP status codes are defined:

- 200: OK, the request was successful; details are included in the response body
- 202: Accepted, e.g. card details have been accepted by Enrolment service, but enrolment is outstanding, dependent upon further checks, identified by response data
- 204: No content, the service completed successfully and there is no content to be returned
- 400: Bad request, see `Error` object for details, e.g. identifies a malformed or invalid request
- 401: Unauthorised, see `Error` object for details, e.g. authorisation token validation failure
- 403: Forbidden, see `Error` object for details, e.g. client identity (origin) not validated
- 404: Not found, see `Error` object for details, e.g. the reference to the SRC Profile in the request data was not found
- 409: Conflict, see `Error` object for details, e.g. the submitted Consumer Identity(s) are already bound to an established SRC Profile
- 500: Internal server error, see `Error` object for details

5.1.2 Error Handling

In case an API service call response contains an HTTP error status code (4xx, 5xx), then the response body contains only an `error`, and the `error` includes details about the error.

5.1.3 Conditionality of Data

Definitions of data conditionality for the APIs are provided based on successful outcomes for those APIs. In case of error outcomes, only an `error` will be returned.

5.1.4 Authorisation

The SRC System uses an authorisation object provided by the API client to identify if there is an existing SRC Profile on which to perform the API operation and determine whether identity validation must occur.

The SRC System supports two categories of authorisation objects:

- Federated ID Token: A Federated ID Token is a digitally signed attestation that the identity of the requestor has been validated by an SRC System. A Federated ID Token issued by one SRC System may be sent by the client to any other participating SRC System
- First Party Token: An opaque first party token issued and recognised by the same SRC System. The content and structure of these tokens is out of scope of the specification

The authorisation objects may be provided by the API client as HTTP header value, e.g. Authorisation, or in the body of HTTP request as explicitly defined by the respective operation, or through other mechanism depending on the integration model and specificity of the individual operation.

5.1.5 Recognition

The SRC System binds device/app identifiers to an SRC Profile to enable the relevant SRC Profile to be determined when device/app identifiers are provided in an API request. Once the relevant SRC Profile has been determined, the SRC System performs additional identity verification if necessary. An SRC System that recognises, and can validate the associated Consumer Identity, returns a Federated ID Token that may be sent by the client to any other participating SRC System.

The Is Recognized API (Section 5.7.4) provides two mechanisms for recognising the device and/or app depending on the specific calling client:

- Implicit cookie-based recognition: the HTTP client (e.g. web browser) may provide a secure HTTP cookie (containing a First Party Token) in the HTTP header that enables the SRC System to identify an SRC Profile

- Explicit token-based recognition: the HTTP client may provide the `appInstanceId` value explicitly in the request body

The SRC System issues a Federated ID Token after successful verification of the tokens described above, and after any necessary additional identity validation.

5.1.6 API Access Control

Access to all APIs must be protected using an authorisation mechanism defined by the SRC System. For server-to-server API access, mutually authenticated TLS connections are generally recommended. For browser-to-server APIs, besides the server authenticated TLS connection, SRC Systems may choose to implement additional access protection models, in order to authenticate that all incoming requests are generated by Onboarded SRC System Participants.

Refer to Annex A Security Guidelines of the SRC Core Specification for more details on the various security credentials used in TLS connections, along with different versions and cipher suites.

5.2 Card Service

Card Service supports Payment Card digitisation. It covers operations to enrol a card, delete a card, to add a billing address card to a previously enrolled card and to retrieve a Digital Card and related masked card data.

5.2.1 Card Enrolment

The Card Enrolment operation enrols a Consumer and Digital Card (associated with an underlying PAN) to a new SRC Profile, or adds a Digital Card to an existing SRC Profile.

If an existing SRC Profile is identified, a Digital Card (associated with an underlying PAN) will be added to that SRC Profile. In the case that an SRC Profile cannot be identified, the SRC System will either create a new SRC Profile based on the Consumer Identity provided, or the Digital Card will be enrolled in an unbounded state. An unbound Digital Card can be bound to an SRC Profile in a subsequent Add Consumer Identities operation, leveraging a first party opaque authorisation token provided in response to this operation.

Table 5.1: Card Enrolment Definition (HTTP with JSON)

HTTP Verb	POST
Path	/cards

Request Body	{
	required String srcClientId;
	conditional String srcDpaId;
	conditional String srcCorrelationId;
	optional String serviceId;
	optional String srciTransactionId;
	conditional Card card;
	conditional JWE<Card> encryptedCard;
	optional JSONObject srcTokenRequestData;
	optional JSONObject threeDsInputData;
	optional AssuranceData assuranceData;
	conditional Consumer consumer;
	conditional JWE<Consumer> encryptedConsumer;
	optional AppInstance appInstance;
	optional DigitalCardData digitalCardData;
	conditional CardholderData cardholderData;
	conditional JWE<CardholderData>
	encryptedCardholderData
	optional ComplianceSettings complianceSettings;
	optional Origin cardSource;
	~~conditional String srcDigitalCardId;~~ **DEPRECATED**
	Replaced by
	conditional EnrollmentReferenceData
	enrollmentReferenceData;
	}
	
	Notes on Conditionality:
	• ~~Exactly one of "card", "srcDigitalCardId", or "encryptedCard" must be provided~~ DEPRECATED
	• Exactly one of `card`, `encryptedCard` or `enrollmentReferenceData` must be provided
	• Either none or one of `consumer` or `encryptedConsumer` can be provided.
	• Either none or one of `cardholderData` or `encryptedCardholderData` can be provided.
	• `srcDpaId` must be provided except when the calling client is an SRCPI
	• `srcCorrelationId` must be provided if available within the present checkout session (e.g. received in an earlier response during the present session), otherwise a new checkout session will be initiated

Response Headers	N/A

Response Body	In case the operation is processed successfully:
	`{`
	`optional String srcCorrelationId;`
	`required MaskedCard maskedCard;`
	`conditional MaskedConsumer maskedConsumer;`
	`conditional string authorization;`
	`conditional string appInstanceId;`
	`}`
	Notes on Conditionality:
	If the request is processed successfully and the card is enrolled into the SRC System, the SRC System will respond with an HTTP 200 status code. In this case:
	• `maskedCard` has to be present
	• `maskedConsumer` will only be provided if `consumer` was provided in the request
	• `authorization` will only be provided if no `authorization` was provided in the request
	If the request is processed successfully, but the card is pending further checks or authentication must be performed before enrolment can be completed, then the service will respond with an HTTP 202 status code, with the same response body as per HTTP 200. Specifically, there are three cases to consider:
	• Address Verification Service (AVS): In this case, the consumer should be prompted to provide billing address details to support the pending AVS check for card enrolment
	• Strong Customer Authentication (SCA): In this case, consumer should be redirected to proceed with a strong authentication mechanism i.e. 3DS non-payment authentication
	• Identity & Verification (ID&V): In this case, the consumer should be taken to the flow performing required identification and verification

HTTP Status Codes	• 200: OK, enrolled card details included in the response body • 202: Accepted, card details have been accepted but enrolment is outstanding, dependent upon further checks, e.g. AVS, requiring the subsequent submission of the card billing address or SCA, requiring 3DS non-payment authentication flow or required ID&V

	• 400: Bad request, see `error` for details. Identifies a malformed or invalid request • 401: Unauthorised, see `error` for details, e.g. authorisation token validation failure • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 500: Internal server error, see `error` for details

5.2.2 Delete Card

The Delete Card operation deletes a Digital Card from an SRC Profile.

If the parameter cardId is a Payment Card Identifier provided by the SRCPI, the relationship of that identifier to Digital Cards is SRC System specific.

Table 5.2: Delete Card Definition (HTTP with JSON)

HTTP Verb	DELETE		
Path	/cards/{cardId}		
Parameters	cardId: Value: may be `srcDigitalCardId` or `srcPaymentCardId`, Required		
Query Parameters	**srcClientId** Type: String	R	Identifies the connecting client, e.g. SRCI, DCF, SRCPI
	srcDpaId Type: String	C	Must be provided except when the calling client is an SRCPI
	srcCorrelationId Type: String	C	If available within the present checkout session (e.g. received in an earlier response during the present session), then it must be provided, otherwise a new checkout session will be initiated
	serviceId Type: String	O	
	srciTransactionId Type: String	O	

	requestor Type: Origin	O	Indicates the original entity requesting deletion of the card from the SRC Profile. Note: the requestor may be different than the API client identified by the `srcClientId`
	reason Type: Card DeletionReason	O	Reason of the card deletion request
Request Body	N/A		
Response Headers	N/A		

Response Body	In case the operation is processed successfully:
	`{`
	`  optional String srcCorrelationId;`
	`}`		

HTTP Status Codes	• 200: OK, card deletion was successful • 400: Bad request, see `error` for details. Identifies a malformed or invalid request • 401: Unauthorised, see error object for details, e.g. authorisation token validation failure • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 404: Not found, see `error` for details, e.g. content of cardId not recognised • 500: Internal server error, see `error` for details		

5.2.3 Add Billing Address

The Add Billing Address operation adds a billing address to an SRC Profile.

Table 5.3: Add Billing Address Definition (HTTP with JSON)

HTTP Verb	POST
Path	/cards/{cardId}/address
Parameters	cardId: Value: `srcDigitalCardId` or `srcPaymentCardId`, Required

Request Body	`{`
	`  required String srcClientId;`
	`  conditional String srcDpaId;`
	`  conditional String srcCorrelationId;`
	`  optional String serviceId;`
	`  optional String srciTransactionId;`
	`  required Address billingAddress;`
	`  optional Boolean setAsShippingAddress;`
	`}`
	Notes on Conditionality:
	• `srcDpaId` must be provided except when the calling client is an SRCPI
	• `srcCorrelationId` must be provided if available within the present checkout session (e.g. received in an earlier response during the present session), otherwise a new checkout session will be initiated

Response Headers	N/A

Response Body	In case the operation is processed successfully:
	`{`
	`  optional String srcCorrelationId;`
	`  required MaskedCard maskedCard;`
	`  conditional MaskedAddress maskedShippingAddress;`
	`}`
	Notes on Conditionality:
	• `maskedShippingAddress` will be provided if `setAsShippingAddress` in the request was set to `true`

HTTP Status Codes	• 200: OK, updated masked card details included in the response body • 400: Bad request, see `error` for details. Identifies a malformed or invalid request • 401: Unauthorised, see `error` for details, e.g. authorisation token validation failure • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 404: Not found, see `error` for details, e.g. the digital card referenced in the request data was not found • 500: Internal server error, see `error` for details

5.2.4 Get Card Data

The Get Card Data operation allows an SRC Participant to retrieve a Digital Card and related masked card data.

Table 5.4: Get Card Data Definition (HTTP with JSON)

HTTP Verb	GET		
Path	/cards/{cardId}		
Parameters	cardId: Value: `srcDigitalCardId`, Required		
Query Parameters	**srcClientId** Type: String	R	Identifies the connecting client, e.g. SRCI, DCF, SRCPI
	srcDpaId Type: String	C	Must be provided except when the calling client is an SRCPI
	srcCorrelationId Type: String	C	If available within the present checkout session (e.g. received in an earlier response during the present session), then it must be provided, otherwise a new checkout session will be initiated
	serviceId Type: String	O	Service identifier associated to an SRC System specific configuration

	srciTransactionId Type: String	O	Identifier of the SRCI transaction
Request Body	N/A		
Response Headers	N/A		

Response Body	In case the operation is processed successfully:
	`{`
	`  required MaskedCard maskedCard;`
	`  optional String srcCorrelationId;`
	`}`		

HTTP Status Codes	• 200: OK, masked card meta-data included in the response body • 400: Bad request, see `error` for details • 401: Unauthorised, see `error` for details, e.g. authorisation token validation failure • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 500: Internal server error, see `error` for details		

5.3 Address Service

The Address Service enables the management of shipping addresses.

5.3.1 Add Shipping Address

The Add Shipping Address operation adds a shipping address to an SRC Profile.

Table 5.5: Add Shipping Address Definition (HTTP with JSON)

HTTP Verb	POST
Path	/addresses

Request Body	{
	required String srcClientId;
	conditional String srcDpaId;
	conditional String srcCorrelationId;
	optional String serviceId;
	optional String srciTransactionId;
	required Address shippingAddress;
	}
	Notes on Conditionality:
	• srcDpaId must be provided except when the calling client is an SRCPI
	• srcCorrelationId must be provided if available within the present checkout session (e.g. received in an earlier response during the present session), otherwise a new checkout session will be initiated

Response Headers	N/A

Response Body	In case the operation is processed successfully:
	{
	optional String srcCorrelationId;
	required MaskedAddress maskedShippingAddress;
	}

HTTP Status Codes	• 200: OK, updated masked card details included in the response body • 400: Bad request, see error for details. Identifies a malformed or invalid request • 401: Unauthorised, see error for details, e.g. authorisation token validation failure • 403: Forbidden, see error for details, e.g. client identity (origin) not validated • 500: Internal server error, see error for details

5.3.2 Delete Shipping Address

The Delete Shipping Address operation deletes a shipping address from an SRC Profile.

Table 5.6: Delete Shipping Address Definition (HTTP with JSON)

HTTP Verb	DELETE

<table>
<tr><td>Path</td><td colspan="3">/addresses/{addressId}</td></tr>
<tr><td>Parameters</td><td colspan="3">addressId: <code>addressId</code> of shipping address to be deleted, Required</td></tr>
<tr><td rowspan="5">Query Parameters</td><td>srcClientId
Type: String</td><td>R</td><td>Identifies the connecting client, e.g. SRCI, DCF, SRCPI</td></tr>
<tr><td>srcDpaId
Type: String</td><td>C</td><td>Must be provided except when the calling client is an SRCPI</td></tr>
<tr><td>srcCorrelationId
Type: String</td><td>C</td><td>If available within the present checkout session (e.g. received in an earlier response during the present session), then it must be provided, otherwise a new checkout session will be initiated</td></tr>
<tr><td>serviceId
Type: String</td><td>O</td><td></td></tr>
<tr><td>srciTransactionId
Type: String</td><td>O</td><td></td></tr>
<tr><td>Request Body</td><td colspan="3">N/A</td></tr>
<tr><td>Response Headers</td><td colspan="3">N/A</td></tr>
<tr><td>Response Body</td><td colspan="3">In case the operation is processed successfully:
<pre>{
    optional String srcCorrelationId;
}</pre></td></tr>
<tr><td>HTTP Status Codes</td><td colspan="3">• 200: OK, the service completed successfully
• 400: Bad request, see <code>error</code> for details. Identifies a malformed or invalid request
• 401: Unauthorised, see <code>error</code> for details, e.g. authorisation token validation failure
• 403: Forbidden, see <code>error</code> for details, e.g. client identity (origin) not validated</td></tr>
</table>

	• 404: Not found, see `error` for details, e.g. `addressId` not recognised • 500: Internal server error, see `error` for details

5.4 SRC Profile Service

The SRC Profile Service enables SRC System Participants to retrieve SRC Profiles from SRC Systems and manage binding of identities to SRC Profiles.

5.4.1 Prepare SRC Profile

The Prepare SRC Profile operation requests that an SRC System prepare one or more SRC Profile(s) to be returned.

Table 5.7: Prepare SRC Profile Definition (HTTP with JSON)

HTTP Verb	POST
Path	/profiles/prepare

Request Body	`{`
	`  required String srcClientId;`
	`  conditional String srcDpaId;`
	`  conditional String srcCorrelationId;`
	`  optional String serviceId;`
	`  optional String srciTransactionId;`
	`  conditional List<JWT> idTokens;`
	`  conditional List<ConsumerIdentity>`
	`  consumerIdentities;`
	`  optional DpaTransactionOptions`
	`  dpaTransactionOptions;`
	`  optional DpaData dpaData;`
	`}`
	Notes on Conditionality:
	• `srcDpaId` must be provided except when the calling client is an SRCPI
	• `srcCorrelation Id` must be provided if available within the present checkout session (e.g. received in an earlier response during the present session), otherwise a new checkout session will be initiated
	• Either `idTokens` or `consumerIdentities` must be provided if:

	• The `idTokens` list carries one or more Federated ID Tokens. Used to identify associated SRC Profile(s), and attest that the requester is authorised to access this data • The `consumerIdentities` list carries one or more Consumer Identities and is used to identify associated SRC Profile(s). It may be used only when the client is trusted and authorised to access the SRC System
Response Headers	N/A

Response Body	In case the operation is processed successfully:
	`{`
	`optional String srcCorrelationId;`
	`required List<SrcProfile> profiles;`
	`conditional String srcDpaId;`
	`}`
	The `profiles` list will contain entries if one or more SRC Profiles are found. Otherwise an empty `profiles` list should be returned.
	Notes on Conditionality:
	• `srcDpaId` must be provided if the DPA Registration occurred based on the `dpaData` in the request

HTTP Status Codes	• 200: OK, SRC Profile details included in the response body • 400: Bad request, see `error` for details. Identifies a malformed or invalid request, including reporting that the `srcCorrelationId` provided was invalid or not recognised • 401: Unauthorised, see `error` for details, e.g. token validation failure • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 500: Internal server error, see `error` for details

5.4.2 Add Consumer Identities

The Add Consumer Identities operation binds a Device Identity (an application instance) or Consumer Identity to an SRC Profile.

In the case that the SRC Profile cannot be located, the SRC System may create a new SRC Profile (based on Consumer details provided in the request) if a previously enrolled unbound Digital Card exists.

The Add Consumer Identifiers operation supports Consumer Identities such as e-mail address, phone number and/or application instance information to support a range of use-cases.

When the type of a provided Consumer Identity is considered to be a primary identity for an SRC Profile (e.g. an email address or phone number), then, if the SRC System detects that an SRC Profile already exists with the same primary identity, the SRC System should respond to the request by advising that an SRC Profile with that identity already exists.

Whether or not a provided Consumer Identity is used to replace an existing identity on an existing SRC Profile is an SRC System implementation decision.

Table 5.8: Add Consumer Identities Definition (HTTP with JSON)

HTTP Verb	POST
Path	/profiles

Request Body	`{`
	`required String srcClientId;`
	`conditional String srcDpaId;`
	`conditional String srcCorrelationId;`
	`optional String serviceId;`
	`optional String srciTransactionId;`
	`conditional Consumer consumer;`
	`conditional AppInstance appInstance;`
	`optional AssuranceData assuranceData;`
	`optional ComplianceSettings complianceSettings;`
	`conditional String srcDigitalCardId;`
	`}`
	
	Notes on Conditionality:
	• `srcDpaId` must be provided except when the calling client is an SRCPI
	• `srcCorrelationId` must be provided if available within the present checkout session (e.g. received in an earlier response during the present session), otherwise a new checkout session will be initiated
	• One or both of `consumer` or `appInstance` must be provided
	• `srcDigitalCardId` must be provided if the request is to establish a new SRC Profile and bind the identifier(s) to a previously enrolled, unbound card

Response Headers	N/A

Response Body	In case the operation is processed successfully:
	`{`
	`optional String srcCorrelationId;`
	`conditional String authorization;`
	`conditional MaskedConsumer maskedConsumer;`
	`conditional String appInstanceId;`
	`}`
	Notes on Conditionality:
	• `authorization` must be provided if `consumer` was provided in the request and can be provided if `appInstance` was provided in the request
	• `maskedConsumer` must be provided if `consumer` was provided in the request
	• `appInstanceId` must be provided if `appInstance` was provided in the request

HTTP Status Codes	• 200: OK, details about the outcome of the call are included in the response body • 400: Bad request, see `error` for details. Can be used to report that one or more input parameters in the request body was not valid • 401: Unauthorised, see `error` for details, e.g. authorisation token validation failure • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 404: Not found, the SRC Profile was not found • 409: Conflict, the submitted Consumer Identity(s) is already bound to an existing SRC Profile • 500: Internal server error, see `error` for details

5.4.3 Unbind App Instance

The Unbind App Instance operation unbinds a Device Identity (an application instance) from an SRC Profile.

Table 5.9: Unbind App Instance Definition (HTTP with JSON)

HTTP Verb	DELETE

Path	~~/profile/appinstances~~ DEPRECATED ***Replaced by*** /profiles/appinstances		
Parameters	**srcClientId** Type: String	R	Identifies the connecting client, e.g. SRCI, DCF, SRCPI
	srcDpaId Type: String	C	Must be provided except when the calling client is an SRCPI
	srcCorrelationId Type: String	C	If available within the present checkout session (e.g. received in an earlier response during the present session), then it must be provided, otherwise a new checkout session will be initiated
	serviceId Type: String	O	
	srciTransactionId Type: String	O	
	appInstanceId Type: String	O	A unique identifier of an app/device issued by the given SRC System. See details in Section 5.1.5 Recognition
Request Body	N/A		
Response Headers	N/A		

Response Body	In case the operation is processed successfully:
	`{`
	`  optional String srcCorrelationId;`
	`}`		

HTTP Status Codes	• 200: OK, the service completed successfully • 401: Unauthorised, see `error` for details, e.g. authorisation token validation failure		

	• 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 404: Not found, can be used to report that the SRC Profile referenced by the authorisation token provided was not found • 500: Internal server error, see `error` for details

5.5 Checkout Service

The Checkout Service provides Payment Data and payment related data for a specific checkout. It also allows provisioning of transaction credentials and retrieval or delivery of the `encryptedPayload` or `encryptedSignedPayload` to support a wide range of checkout use cases.

5.5.1 Prepare Checkout Data

The Prepare Checkout Data operation allows the SRCI to create a checkout request to fetch the DCF information along with the SRC checkout request JWS for the DCF.

The resulting `checkoutRequestJws` is signed by the SRC System and this structure needs to be passed to the SRC System for Checkout operation.

Table 5.10: Prepare Checkout Data Definition (HTTP with JSON)

HTTP Verb	POST
Path	/transaction/preparedata
Request Body	{ required String srcClientId; conditional String srcDpaId; conditional String srcCorrelationId; optional String serviceId; optional String srciTransactionId; required String srcInitiatorId; optional PayloadTypeIndicator payloadTypeIndicatorCheckout; optional PayloadTypeIndicator payloadTypeIndicatorPayload; optional String recipientIdCheckout; optional String recipientIdPayload; optional JSONObject customInputData; required String srcDigitalCardId;

	~~conditional String consumerId;~~DEPRECATED ***Replaced by*** conditional String srcConsumerId; optional List<String> shippingAddressIds; optional String authorization; required DpaTransactionOptions dpaTransactionOptions; optional DpaData dpaData; optional AssuranceData assuranceData; optional String checkoutResponseUri; } **Notes on Conditionality:** • srcDpaId must be provided except when the calling client is an SRCPI • srcCorrelationId must be provided if available within the present checkout session (e.g. received in an earlier response during the present session), otherwise a new checkout session will be initiated • ~~consumerId : if available within the present checkout session (e.g. received in an earlier API response during the present session), then it must be provided~~ DEPRECATED • srcConsumerId must be provided if available within the present checkout session (e.g. received in an earlier API response during the present session),
Response Headers	N/A

Response Body	In case the operation is processed successfully:
	{
	required String srcCorrelationId;
	required JWS<CheckoutRequest> checkoutRequestJws;
	}
	
	The definition of checkoutRequestJws included in Section 2.4.1 Checkout Request JWS.

HTTP Status Codes	• 200: OK, details about the outcome of the call are included in the response body • 400: Bad request, see error for details. Can be used to report that one or more input parameters in the request body was not valid

	• 401: Unauthorised, see `error` for details, e.g. authorisation token validation failure • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 404: Not Found, used to indicate the checkout flow does not require redirection to a DCF and the Checkout operation can be performed instead • 500: Internal server error, see `error` for details

5.5.2 Checkout

The Checkout operation utilises the Consumer's chosen Digital Card and details of the current transaction to retrieve Payment Data and payment related data.

If present in the `checkoutResponse` attribute, the `encryptedPayload` or `encryptedSignedPayload` encrypted according to JSON Web Encryption (JWE) specification RFC 7516 and the algorithm used for encryption is according to RFC 7518 Section 4.1.

Table 5.11: Checkout Definition (HTTP with JSON)

HTTP Verb	POST
Path	/transaction/credentials

Request Body	For requests containing the signed `checkoutRequestJws` object, the request body shall contain the following:
	`{`
	`  required String srcClientId;`
	`  conditional String srcDpaId;`
	`  conditional String srcCorrelationId;`
	`  optional String serviceId;`
	`  optional String srciTransactionId;`
	`  optional String shippingAddressId;`
	`  optional Address shippingAddress;`
	`  optional AcceptanceChannelRelatedData`
	`  acceptanceChannelRelatedData;`
	`  optional ComplianceSettings complianceSettings;`
	`  required JWS<CheckoutRequest> checkoutRequestJws;`
	`}`

	Alternatively, for requests containing only unsigned checkout request data, then the request body shall contain the following:
	
	`{`
	`required String srcClientId;`
	`conditional String srcDpaId;`
	`conditional String srcCorrelationId;`
	`optional String serviceId;`
	`optional String srciTransactionId;`
	`optional PayloadTypeIndicator payloadTypeIndicatorCheckout;`
	`optional PayloadTypeIndicator payloadTypeIndicatorPayload;`
	`optional String recipientIdCheckout;`
	`optional String recipientIdPayload;`
	`required String srcDigitalCardId;`
	`optional String shippingAddressId;`
	`optional Address shippingAddress;`
	`conditional DpaTransactionOptions dpaTransactionOptions;`
	`optional AcceptanceChannelRelatedData acceptanceChannelRelatedData;`
	`conditional DpaData dpaData;`
	`optional AssuranceData assuranceData;`
	`optional ComplianceSettings complianceSettings;`
	`}`
	
	Notes on Conditionality:
	• `srcCorrelationId` must be provided if available within the present checkout session (e.g. received in an earlier response during the present session), otherwise a new checkout session will be initiated
	• `dpaTransactionOptions` must be provided if 3DS is to be performed by SRC System, or if default configuration values are required to be overridden for a given transaction, or if the calculation of Dynamic Data is dependent on knowing the transaction amount
	• Either `srcDpaId` or `dpaData` must be provided by the client except when the calling client is an SRCPI.

Response Headers	N/A

Response Body	In case the operation is processed successfully:
	`{`
	`required JWS<CheckoutPayloadResponse> checkoutResponse`

	} **Additional Notes:** • Presence of the `encryptedPayload` or `encryptedSignedPayload` within the `checkoutResponse` (see Table 2.46) depends on the value of `payloadTypeIndicatorCheckout` parameter (as dynamically supplied in the request (query) or statically derived using the default configured during DPA Registration) being set to any valid value other than SUMMARY
HTTP Status Codes	• 200: OK, transaction credential response details included in the response body • 400: Bad request, see `error` for details • 401: Unauthorised, see `error` for details, e.g. authorisation token validation failure • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 500: Internal server error, see `error` for details

5.5.3 Get Payload

The Get Payload operation returns Payment Data and payment related data to be used in payment authorisation.

The Get Payload operation is a server-side API intended for server-based communication.

Table 5.12: Get Payload Definition (HTTP with JSON)

HTTP Verb	GET		
Path	/transaction/credentials		
Parameters	**srcClientId** Type: String	R	Identifies the connecting client, e.g. SRCI, DCF, SRCPI
	payloadTypeIndicator Type: PayloadTypeIndicator	O	Identifies the type of encrypted payload to be returned. A value of SUMMARY is invalid for the Get Payload operation

	recipientId Type: String	O	Identifies the recipient of the encrypted payload known to the SRC System. The SRC System will use this value to determine the key used for encryption of the payload
	srcDpaId Type: String	C	Must be provided except when the calling client is an SRCPI
	srcCorrelationId Type: String	R	Reference to the checkout session
	serviceId Type: String	O	Service identifier associated to an SRC System specific configuration
	srciTransactionId Type: String	O	
Request Body	N/A		
Response Headers	N/A		

Response Body	In case the operation is processed successfully:
	`{`
	`required JWS<CheckoutPayloadResponse> payloadResponse;`
	`}`
	When `payloadTypeIndicator` in this request or `payloadTypeIndicatorPayload` in the Checkout operation or checkout() method is set to SUMMARY this request should return an HTTP Status Code of 400 indicating an invalid request.
	Additional Notes:
	• Presence of either the `encryptedPayload` or `encryptedSignedPayload` data element within the `payloadResponse` (see Table 2.46) is always required		

HTTP Status Codes	• 200: OK, transaction credential response details included in the response body • 400: Bad request, see `error` for details • 401: Unauthorised, see `error` for details, e.g. authorisation token validation failure • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 500: Internal server error, see `error` for details

5.5.4 Make Payment

The Make Payment operation allows an SRC System to send payload information for authorisation purposes directly to a payment SRCI.

Table 5.13: Make Payment Definition (HTTP with JSON)

HTTP Verb	POST
Path	/transaction/credentials

Request Body	`{`
	`  required JWS<CheckoutPayloadResponse>`
	`  signedTransactionCredentials;`
	`}`

Response Headers	N/A
Response Body	NA
HTTP Status Codes	• 204: Accepted. No additional content • 400: Bad request, see `error` for details. Can be used to report that one or more input parameters in the request body was not valid • 401: Unauthorized, see `error` for details, e.g. authorisation token validation failure • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 500: Internal server error, see `error` for details

5.6 Confirmation Service

The Confirmation Service enables SRC Participants to notify the SRC System of the checkout or payment results.

5.6.1 Confirmation

The Confirmation operation enables SRC Participants to provide a notification of the result of a checkout service (checkout or payment authorisation).

The Confirmation operation is server-side API intended for server-based communication.

Table 5.14: Confirmation Definition (HTTP with JSON)

HTTP Verb	POST
Path	/confirmations

Request Body	{
	required String srcClientId;
	conditional String srcDpaId;
	required String srcCorrelationId;
	optional String serviceId;
	optional String srciTransactionId;
	optional AssuranceData assuranceData;
	optional JSONObject customData;
	~~required ConfirmationData confirmationData;~~
	DEPRECATED
	Replaced by
	Required ConfirmationData2 confirmationData2;
	}
	
	Notes on Conditionality:
	• `srcDpaId`: must be provided except when the calling client is an SRCPI

Response Headers	N/A
Response Body	N/A
HTTP Status Codes	• 200: OK • 204: No content, the confirmation message was accepted • 400: Bad request, see `error` for details

	• 401: Unauthorised, see `error` for details, e.g. authorisation token validation failure • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 500: Internal server error, see `error` for details

5.7 Identity Service

The Identity Service enables operations related to identity recognition, validation of identity and the generation of Federated ID Tokens.

The service allows identity validation to be a two-step process encompassing initiation and completion to allow challenge/response interaction with the Consumer within the SRC experience. It is also possible that an out of band mechanism be used in which case the challenge/response within the SRC experience may not be necessary.

When requested, the SRC System should perform the validation of the identity (to verify possession) regardless of whether the Consumer Identity is associated with an SRC Profile or not.

5.7.1 Identity Lookup

The Identity Lookup operation utilises a provided Consumer Identity (email address or mobile phone number) to determine whether it is associated with an SRC Profile.

Table 5.15: Identity Lookup Definition (HTTP with JSON)

HTTP Verb	POST
Path	/identities/lookup

Request Body	{
	required String srcClientId;
	optional String serviceId;
	required ConsumerIdentity consumerIdentity;
	}

Response Body	In case the operation is processed successfully: { ~~conditional Boolean consumerPresent;~~DEPRECATED ***Replaced by*** required Boolean consumerPresent1;

	`conditional ConsumerStatus consumerStatus;`
	`conditional String idLookupSessionId;`
	`conditional List<IdentityValidationChannel>`
	`supportedValidationChannels;`
	`}`
	Notes on Conditionality:
	• ~~`consumerPresent` must be provided if the specified Consumer Identity was recognised by the SRC System~~ DEPRECATED
	• `consumerStatus, idLookupSessionId` and the list of `supportedValidationChannels` must all be provided if the specified Consumer Identity was recognised by the SRC System

HTTP Status Codes	• 200: OK, lookup result included in the response body • 400: Bad request, see `error` for details • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 500: Internal server error, see `error` for details

5.7.2 Initiate Identity Validation

The Initiate Identity Validation operation initiates a process to validate that a Consumer is in the possession of, or has access to, the Consumer Identity claimed.

Table 5.16: Initiate Identity Validation Definition (HTTP with JSON)

HTTP Verb	POST
Path	/identities/validation/initiate

Request Body	`{`
	`required String srcClientId;`
	`optional String serviceId;`
	`conditional ConsumerIdentity consumerIdentity;`
	`conditional String idLookupSessionId;`
	`optional IdentityValidationChannel`
	`requestedValidationChannel;`
	`}`
	Notes on Conditionality:
	• Either the `consumerIdentity` object or the `idLookupSessionId` value needs to be provided, but not both

Response Body	In case the operation is processed successfully:
	`{`
	`required String idValidationSessionId;`
	`required IdentityValidationChannel maskedValidationChannel;`
	`optional String validationMessage;`
	`optional List<IdentityValidationChannel> supportedValidationChannels;`
	`}`

HTTP Status Codes	• 200: OK, verification request results included in the response body • 400: Bad request, see `error` for details • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 404: Not found, conveys that the identity provided was not recognised • 500: Internal server error, see `error` for details

5.7.3 Complete Identity Validation

The Complete Identity Validation operation determines whether data, provided by the Consumer as part of a second step of an identity validation process, is valid. It can also be used to check whether an out-of-band service was successful.

Table 5.17: Complete Identity Validation Definition (HTTP with JSON)

HTTP Verb	POST
Path	/identities/validation/complete

Request Body	{
	`required String srcClientId;`
	`optional String serviceId;`
	`required String idValidationSessionId;`
	`conditional String validationData;`
	}
	Notes of Conditionality:
	• The `validationData` should be provided if type of identity validation channel was other than OUT_OF_BAND.

Response Body	In case the operation is processed successfully:
	`{`
	`  required JWT idToken;`
	`}`

Response Headers	• Retry-After: may be specified by the server when HTTP status code is 202.
HTTP Status Codes	• 200: OK, validation of identity successfully completed, Federated ID Token included in the response body • 202: OK, validation still in progress and no result yet available. • 400: Bad request, see `error` for details. Identifies a malformed or invalid request • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 500: Internal server error, see `error` for details

5.7.4 Is Recognized

The Is Recognized operation uses a Device Identity (derived from a First Party Token) to determine whether it is bound to an SRC Profile and, if so, returns a Federated ID Token.

Table 5.18: Is Recognized Definition (HTTP with JSON)

HTTP Verb	GET		
Path	/identities/recognize		
Parameters	**srcClientId** Type: String	R	Identifies the connecting client, e.g. SRCI, DCF, SRCPI
	srcDpaId Type: String	C	Must be provided except when the calling client is an SRCPI
	serviceId Type: String	O	
	srciTransactionId Type: String	O	

	appInstanceId Type: String	O	A unique identifier of an app/device issued by the given SRC System. See details in Section 5.1.5 Recognition
Request Body	N/A		

Response Body	In case the operation is processed successfully:
	`{`
	`optional String srcCorrelationId;`
	`required List<JWT> idTokens;`
	~~`conditional String appInstanceId;`~~`DEPRECATED`
	`Replaced by`
	`optional String updatedAppInstanceId;`
	`}`
	Notes on Conditionality:
	• ~~The `appInstanceId` parameter must be supplied if the connecting consumer application instance is recognised by the SRC System~~ DEPRECATED
	• A list of `idTokens` must be provided, one for each established SRC Profile associated to the recognised consumer application instance. Each token should be a Federated ID Token		

HTTP Status Codes	• 200: OK, consumer application instance was recognised, with recognition data included in the response body • 400: Bad request, see `error` for details. Identifies a malformed or invalid request • 403: Forbidden, see `error` for details, e.g. client identity (origin) not validated • 404: Not found, see `error` for details, e.g. unable to locate SRC Profile • 500: Internal server error, see `error` for details		

5.8 Public Keys Retrieval Service

The Public Keys Retrieval service enables retrieval of cryptographic public keys from a well-known URL hosted by an SRC System. The keys retrieved are used by other SRC Participants for Federated ID Token and JWS signature verification.

Each SRC System must host cryptographic public keys for retrieval by other SRC Systems and SRC Participants to allow signature verification and encryption in the following cases:

- Federated ID Token is signed JWT in the form of JWS
- `checkoutRequest`, `checkoutResponse`, `payloadResponse` and (optionally) `encryptedSignedPayload` are signed in the form of JWS
- Payment Card and Consumer details presented during Enrolment can be encrypted in the form of JWE

Each SRC System must publish the cryptographic public keys on the web in well-known location to allow discovery of the keys by the relying party. Each key must be easily identifiable so it can be selected by the relying party based on the key ID ("kid") specified in the header of the JWS.

For signature verification, key retrieval and selection process for SRC Systems follows the steps below:

1. The relying party discovers the URI of the signature issuer by examining the JWS content (i.e. "iss") or using some other method.
2. The relying party retrieves the set of public keys available at the well-known path on issuer host as per issuer URI
3. The relying party examines JWS header to discover the key ID ("kid" member) and cryptographic signature algorithm ("alg" member).
4. The relying party selects the corresponding public key that matched the key ID and performs verification of the signature following the algorithm

For encryption, the recipient party should fetch the key from the well-known path based on a pre-agreed key ID.

Note: *Symmetric Key Retrieval is not defined in this version of the specification.*

5.8.1 Public Key Retrieval

The Public Key Retrieval operation retrieves a set of public keys.

Table 5.19: Public Key Retrieval Definition (HTTP with JSON)

HTTP Verb	GET
Path	/keys

Request Body	N/A
Response Body	JWK Keyset as specified by JSON Web Key standard (RFC 7517). The keyset must specify at least one valid public key. Each key in the keyset must contain the following details: • Key ID (“kid”) used for key selection as described in the flow above • Key type (“kty”). It is also recommended to specify Key Operations (“key_ops”) with value “verify” to indicate the public key intended use. • The key is specified as an X.509 certificate chain (“x5c”)
HTTP Status Codes	Standard HTTP error codes.

Handling of keys used to encrypt the `encryptedPayload` or the `encryptedSignedPayload` returned by SRC Systems is out the scope of the SRC Specifications. The encryption algorithms and keys should be specified by SRC Programme.

6 Notification Service

The Notification service enables outbound messages sent by the SRC System when specific events occur.

6.1 Notifications Principles

The notifications are sent as HTTP POST messages to the specific endpoint. The SRC System must support HTTPS and use it as default.

The SRC System maintains a registry of the SRC Participants (notification subscribers) for any given event. Each notification subscriber must be Onboarded to the SRC System and the base URL of the notification subscriber's server provided as configuration data. The Onboarding and configuration of the notification subscribers are out of scope of this document.

Each notification defines a specific path that should be appended to the base URL specified for the notification subscriber.

The "Success" HTTP status code indicates to the SRC System that the notification has been received, acknowledged and understood. In case of an error, client or server, the entity containing explanation of error condition should be provided.

6.1.1 Data Delivery Modes

Where applicable, the following two data delivery models should be considered:

- Push Model – where the SRC System includes the data in the body of the notification. The notification subscriber receives the full set of data associated with the event that triggered the notification
- Push-Pull Model – where the SRC System only includes a specific entity identifier or session identifier (and optionally a First Party Token) in the request body of the notification. The notification subscriber willing to act on the notification received should refer to the specific API to fetch the data associated with the event that triggered the notification

The SRC System may support either one or both data delivery models.

6.1.2 Standard HTTP Status Codes

For the notifications the standard classes of HTTP status codes should be used by the server hosting subscriber's notification endpoint. These are described in Table 6.1.

Table 6.1: Standard HTTP Status Codes

Code Class	Type	Description
2XX	Success	This class of status codes indicates the notification was received by the subscriber, understood, accepted.
3XX	Redirection	Indicates that further action may be taken by the SRC System in order to fulfil the delivery of notification. SRC System is under no obligation to follow the actions indicated.
4XX	Client Error	Intended for cases in which the SRC System originating the notification seems to have encounter an error and therefore the subscriber's endpoint cannot acknowledge the reception of the notification.
5XX	Server Error	Indicate cases in which the subscriber's server is aware that it has encountered an error or is otherwise incapable of handling the notification.

In case of the error HTTP codes, the subscriber's server should include a standard Error entity containing an explanation of the error situation.

Support for individual HTTP codes for the classes given above is optional for the SRC System.

6.2 Card Update Event Notification

The Card Update Event notification sends a message to subscribers when a Digital Card's information has been modified or updated.

Each notification must specify the timestamp of the event and must contain the reason for the modification or update.

The SRC System may support two notification delivery models:

- The request body may contain the `maskedCard` object representing the updated Digital Card, *or*
- The request body may only contain the `srcDigitalCardId` along with the optional `authorization` (a First Party Token). The subscriber may then fetch the `maskedCard` object using the Get Card Data operation

Table 6.2: Card Update Notification Definition (HTTP with JSON)

HTTP Verb	POST
Path	/notifications/cards

Request Body	`{`
	`  required List<DigitalCardUpdateNotification>`
	`  digitalCardUpdateNotifications;`
	`}`

Response Headers	N/A
Response Body	N/A
HTTP Status Codes	Standard classes of HTTP status codes apply. Specifically, the following individual codes may be used: • 204: No Content, the subscriber acknowledges the receipt of the notification • 400: Bad Request, the request body has been malformed or otherwise prohibits the subscriber to process the notification

6.3 Identity Validation Completion Event Notification

The Identity Validation Complete Event notification sends a message to subscribers when an SRC System determines, or is itself notified, that an out-of-band identity validation service has completed.

The SRC System may support two notification delivery models:

- The request body may contain a Federated ID Token; *or*
- The request body may contain an `idValidationSessionId` along with the optional `authorization` (a First Party Token). The subscriber may the fetch the Federated ID Token using the Complete Identity Validation operation

Table 6.3: Complete Identity Validation Notification Definition (HTTP with JSON)

HTTP Verb	POST

Path	/notifications//identities/validation/complete

Request Body	{
	required String idValidationSessionId;
	conditional JWT idToken;
	conditional Error error;
	optional String authorization;
	}
	
	Notes of Conditionality:
	• When the subscriber is configured in the push model, then either `idToken` or `error` should be present.

Response Headers	N/A
Response Body	N/A
HTTP Status Codes	Standard classes of HTTP status codes apply. Specifically, the following individual codes may be used: • 204: No Content, the subscriber acknowledges the receipt of the notification • 400: Bad Request, the request body has been malformed or otherwise prohibits the subscriber to process the notification

6.4 Payment Notification

The Payment Notification sends a message to subscribers when an SRC System has received a confirmation of payment authorisation.

Each Payment Notification must specify the timestamp of the payment completion event and must contain the status of the payment authorisation.

Table 6.4: Payment Notification Definition (HTTP with JSON)

HTTP Verb	POST
Path	/notifications/payment

Request Body	{
	required String srcCorrelationId;
	optional String srciTransactionId;
	optional String serviceId;
	optional String srcDigitalCardId;
	required ConfirmationData2 confirmationData2;
	optional JSONObject customData;
	}
	}

Response Headers	N/A
Response Body	N/A
HTTP Status Codes	Standard classes of HTTP status codes apply. Specifically, the following individual codes may be used: • 204: No Content, the subscriber acknowledges the receipt of the notification • 400: Bad Request, the request body has been malformed or otherwise prohibits the subscriber to process the notification

Annex A EMVCo Specification Mapping

This Annex describes the mapping of data from a non-SRC EMVCo Specification to these SRC Specifications. It provides a level of interoperability for an implementation using one non-SRC EMVCo specification to use SRC to process such a transaction.

A.1 Merchant-Presented Mode – QR Code Payload

Annex A.1 describes the mapping of data from the QR Code Payload described in the Merchant-Presented Mode specification (EMV® QR Code Specification for Payment Systems (EMV QRCPS) – Merchant-Presented Mode). As per the EMV Merchant Presented Mode specification, this describes any conversion of data necessary as well as any additional data needed to process the transaction.

The mapping is described from the perspective of the Mobile Application consuming a QR Code Payload and building the SRC data elements to be populated to SRC API input parameters and SRC JavaScript SDK attributes. The descriptions below all assume that the QR Code Payload complies with the EMV Merchant Presented Mode specification.

A.1.1 SRC Data Elements

The following SRC data elements, parameters, objects or attributes are populated with Merchant Presented Mode QR specific data. Based on the mapping described, when processing a Merchant-Presented QR Code payment transaction using SRC, `acceptanceChannelRelatedData` is a required input parameter or attribute in the relevant SRC API operation or SRC JavaScript SDK method.

Transaction Amount

The `transactionAmount` data element of the `TransactionAmount` object is populated with:

- The transaction amount; *or*
- When there is also a tip or convenience fee (see Annex A.1.4 QR Code Specific Data Elements for Additional Amounts), the sum of the transaction amount and the tip or convenience fee

The value of the transaction amount is dependent on the presence of the Transaction Amount (ID “54”) in the QR Code Payload.

- If present, the transaction amount referred to above is the value present in Transaction Amount (ID “54”) in the QR Code Payload

- If not present, the transaction amount referred to above is the Consumer-entered amount

Transaction Currency

The `transactionCurrency` data element of the `TransactionAmount` object is populated with the Transaction Currency (ID "53") in the QR Code Payload.

Acceptance Channel Type

The `acceptanceChannelType` data element of the `AcceptanceChannelRelatedData` object is populated with the value of EMV_MERCHANT_PRESENTED_MODE

Acceptance Channel Technology

The `acceptanceChannelTechnology` data element of the `AcceptanceChannelRelatedData` object is populated with the value of QR_CODE.

Digital Payment Application Data

The `merchantAccountInformation` data element of the `DpaData` object is populated with the Merchant Account Information (IDs "02" to "51") of the QR Code Payload. It is only necessary to populate the content of the ID relevant to the receiving SRC System, which is based on which SRC System maintains the Digital Card selected for the specific transaction:

- ID "02" or "03" for the Visa SRC System
- ID "04" and "05" for the Mastercard SRC System
- ID "09" or "10" for the Discover SRC System
- ID "11" and "12" for the Amex SRC System
- ID "13" or "14" for the JCB SRC System
- ID "15" and "16" for the Union Pay SRC System

A.1.2 QR Code specific Data Elements for Seller Data

The following data element is defined specifically for the `sellerData` object which is a data element of the `AcceptanceChannelData` object (see Table 2.2).

QR Code Payload

Always populated with the full content of the QR Code Payload.

Table A.1: SRC API Usage for QR Code Payload

qrCodePayload	
Type	String
Constraint	Maximum length of 2048
Present in object	`sellerData`

A.1.3 QR Code specific Data Elements for Consumer Data

The following data elements are defined specifically for the `consumerData` object which is a data element of the `AcceptanceChannelData` object (see Table 2.2). Consumer Data will only be present and populated if one or more of the following data objects are indicated within the QR Code Payload

Bill Number

Populated with a Consumer-entered bill number if the Bill Number (ID “01”), with a value of “***”, is present within the Additional Data Field Template (ID “62”) of the QR Code Payload.

Table A.2: SRC API Usage for Bill Number

billNumber	
Type	String
Constraint	Maximum length of 25
Present in object	`consumerData`

Mobile Number

Populated with a Consumer-entered mobile number if the Mobile Number (ID “02”), with a value of “***”, is present within the Additional Data Field Template (ID “62”) of the QR Code Payload.

Table A.3: SRC API Usage for Mobile Number

mobileNumber	
Type	String
Constraint	Maximum length of 25
Present in object	consumerData

Store Label

Populated with a Consumer-entered store label if the Store Label (ID “03”), with a value of “***”, is present within the Additional Data Field Template (ID “62”) of the QR Code Payload.

Table A.4: SRC API Usage for Store Label

storeLabel	
Type	String
Constraint	Maximum length of 25
Present in object	consumerData

Loyalty Number

Populated with a Consumer-entered loyalty number if the Loyalty Number (ID “04”), with a value of “***”, is present within the Additional Data Field Template (ID “62”) of the QR Code Payload.

Table A.5: SRC API Usage for Loyalty Number

loyaltyNumber	
Type	String
Constraint	Maximum length of 25
Present in object	consumerData

Reference Label

Populated with a Consumer-entered reference label if the Reference Label (ID “05”), with a value of “***”, is present within the Additional Data Field Template (ID “62”) of the QR Code Payload.

Table A.6: SRC API Usage for Reference Label

referenceLabel	
Type	String
Constraint	Maximum length of 25
Present in object	consumerData

Customer Label

Populated with a Consumer-entered Customer label if the Customer Label (ID “06”), with a value of “***”, is present within the Additional Data Field Template (ID “62”) of the QR Code Payload.

Table A.7: SRC API Usage for Customer Label

customerLabel	
Type	String
Constraint	Maximum length of 25
Present in object	consumerData

Terminal Label

Populated with a Consumer-entered terminal label if the Terminal Label (ID “07”), with a value of “***”, is present within the Additional Data Field Template (ID “62”) of the QR Code Payload.

Table A.8: SRC API Usage for Terminal Label

terminalLabel	
Type	String
Constraint	Maximum length of 25

terminalLabel	
Present in object	consumerData

Purpose of Transaction

Populated with a Consumer-entered transaction purpose if the Purpose of Transaction (ID "08"), with a value of "***", is present within the Additional Data Field Template (ID "62") of the QR Code Payload.

Table A.9: SRC API Usage for Purpose of Transaction

purposeOfTransaction	
Type	String
Constraint	Maximum length of 25
Present in object	consumerData

Email

Populated with an email known to the Mobile Application if the Additional Consumer Data Request (ID "09"), with a value containing the character "E", is present within the Additional Data Field Template (ID "62") of the QR Code Payload.

Table A.10: SRC API Usage for Email

email	
Type	String
Constraint	Maximum length of 255
Present in object	consumerData

Phone Number

Populated with an mobile number known to the Mobile Application if the Additional Consumer Data Request (ID "09"), with a value containing the character "M", is present within the Additional Data Field Template (ID "62") of the QR Code Payload.

Table A.11: SRC API Usage for Phone Number

phoneNumber	
Type	String
Constraint	A length ranging from 4 to 14
Present in object	`consumerData`

Address

Populated with an address known to the Mobile Application if the Additional Consumer Data Request (ID “09”), with a value containing the character “A”, is present within the Additional Data Field Template (ID “62”) of the QR Code Payload.

Table A.12: SRC API Usage for Address

address	
Type	String
Constraint	Maximum length of 2048
Present in object	`consumerData`

A.1.4 QR Code Specific Data Elements for Additional Amounts

The following data elements are defined specifically for the `additionalAmounts` list which is a data element the `TransactionAmount` object (see Table 2.39).

Tip

A floating-point number only populated with a Consumer-entered tip value if Tip or Convenience Indicator (ID “55”), containing a value of “01”, is present within the QR Data Payload.

If present with the relevant value, populate an entry of the `additionalAmounts` list with the values in Table A.13.

Table A.13: SRC API Usage for Tip

Data Element	Value
`additionalAmountType`	TIP
`additionalAmountValue`	Consumer entered tip value

Convenience Fee

A floating-point number only populated if the Tip or Convenience Indicator (ID “55”), containing a value of “02” or “03”, is present within the QR Data Payload.

If present with the relevant values, populate an entry of the `additionalAmounts` list with the values in Table A.14.

Table A.14: SRC API Usage for Convenience Fee

Data Element	Value
`additionalAmountType`	CONVENIENCE_FEE
`additionalAmountValue`	If the value of ID “55” is: • “02” then populate with the content of the Value of Convenience Fee Fixed (ID “56”) present within the QR Data Payload (converted to a floating-point number) • “03” then populate with a Mobile Application calculated value, equal to a percentage of the Sub Total. The percentage used for the calculation is the Convenience Fee Percentage (ID “57”) value present in the QR Code Payload

Sub Total

A floating-point number only populated if one of the above `tip` or `convenienceFee` data elements is populated.

If a `tip` or `convenienceFee` data elements is populated, then populate an entry of the `additionalAmounts` list with the values in Table A.15.

Table A.15: SRC API Usage for Sub Total

Data Element	Value
`additionalAmountType`	SUB_TOTAL
`additionalAmountValue`	Either the: • Transaction Amount (ID “54”) if present in the QR Code Payload (converted to a floating-point number); *or* • Consumer-entered amount if the Transaction Amount (ID “54”) is not present in the QR Code Payload

***** END OF DOCUMENT *****