Mastercard CIT/MIT Framework

In late 2021, Mastercard revised their MIT framework to include 8 use cases – similar to Visa MIT framework but with some different nuances. Visa released their MIT framework mandate back in 2017. Previously, Mastercard only required COF indicator to comply to their Stored Credential mandate. In EU, the Mastercard Trace ID (transaction ID) was also required for all subsequent transactions.

What are the differences between the existing Visa MIT framework and the new Mastercard MIT framework?

1. Mastercard has 2 types of recurring transactions (subscription vs standing order.)
   a. **Subscription** = fixed value/fixed frequency
      i. **Example** = magazine subscription
   b. **Standing Order** = variable value/fixed frequency
      i. **Example** = Utility bill
2. Mastercard needs to know MIT type at the time of the initial CIT.
   a. **Example**: At the time of purchasing a magazine subscription, the merchant is required to alert Mastercard the intent of the future transactions (subsequent auth reason = 7 subscription) while the cardholder is present at the initial transaction.
3. Mastercard also has different industry practice use cases.
   - Mastercard doesn't support Reauthorization.
   - Mastercard doesn't support Incremental in their MIT framework but does allow incremental authorizations.
   - Mastercard supports Partial Shipment.
4. Mastercard has a use case called Ad-Hoc which Cybersource considers a standard CIT with COF.

Mastercard Initial and Subsequent CIT

Storing a Credential

The initial transaction must be a cardholder present Customer Initiated Transaction (CIT) which indicates that the credential is being stored on file for the first time (whether a zero-dollar authorization or first transaction).

Note: This could be used for "Credential on File" CITs and Industry Practice MITs.

EU: Under the terms of PSD2 this initial transaction must be strongly authenticated using 3DSecure unless it is a Mail-Order or Telephone-Order (MOTO) transaction.

Important: Using this field when the cardholder is NOT present for subsequent transactions might cause declines because the issuer is expecting authentication data associated with the cardholder being present.

- **SCMP API:**
  ○ subsequent_auth_first = Y
  ○ e_commerce_indicator = internet, MOTO or payer auth value

- **Simple Order API:**

° subsequentAuthFirst = true
　　　° ccAuthService_commerceIndicator = internet, MOTO or payer auth value

- **REST:**
　　　° processingInformation.authorizationOptions.initiator.credentialStoredOnFile = true
　　　° processingInformation.commerceIndicator = internet, MOTO or payer authentication

CIT (COF) – Subsequent CIT using a Stored Credential
Consumer agrees to store the credential on file with the merchant for future cardholder-initiated and/or merchant-initiated transactions that may occur from time to time, or consumer chooses to use credential previously placed on file with the merchant when making a purchase.

- **SCMP API:**
　　　° subsequent_auth_stored_credential=Y
　　　° e_commerce_indicator = internet, MOTO or payer authentication

- **Simple Order API:**
　　　° subsequentAuthStoredCredential = true
　　　° ccAuthService_commerceIndicator = internet, MOTO or payer authentication

- **REST:**
　　　° processingInformation.authorizationOptions.initiator.storedCredentialUsed = true
　　　° processingInformation.commerceIndicator = internet, MOTO or payer authentication

Mastercard Initial CIT for Standing Instruction MITs

Merchant will need to indicate what type of MIT the subsequent transactions will be for some of the Standing-Instruction CITs– either installment, subscription, or standing order.

Note: If the reason is unknown for future MITs, the acceptable default outlined by Mastercard would be CIT with COF.

Installment Payment
An CIT Installment payment is the initiating transaction for an instalment payment plan. This type of payment requires a cardholder agreement so that the merchant can initiate a series of fixed amount charges on a fixed schedule with a defined end date for a single purchase.
Note: See Credit Card Service Guide to confirm Installment as a service API fields. The fields listed below are relevant to the MIT framework, additional fields may be required to comply to Installment mandates.

- **SCMP API:**
　　　° subsequent_auth_first = Y
　　　° e_commerce_indicator = internet, MOTO or payer auth value
　　　° subsequent_auth_reason = 9

- **Simple Order API:**
  - ° subsequentAuthFirst = true
  - ° ccAuthService_commerceIndicator = internet, MOTO or payer auth value
  - ° subsequentAuthReason= 9

- **REST:**
  - ° processingInformation.authorizationOptions.initiator.credentialStoredOnFile = true
  - ° processingInformation.commerceIndicator = internet, MOTO or payer auth value
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason = 9

Recurring – Subscription

A CIT made pursuant to an agreement between a cardholder and a merchant, whereby the cardholder authorizes the merchant to store and use the cardholder's account data for a transaction that just occur at **regularly scheduled intervals for a fixed amount**, as specified in the agreement. The subscription agreement may include an allowance for introductory pricing or for price changes to occur from time to time.

- **SCMP API:**
  - ° auth_first_recurring_payment= Y
  - ° subsequent_auth_first = Y
  - ° e_commerce_indicator = internet, MOTO or payer auth value
  - ° subsequent_auth_reason = 7

- **Simple Order API:**
  - ° firstRecurringPayment=Y
  - ° subsequentAuthFirst = true
  - ° ccAuthService_commerceIndicator = internet, MOTO or payer auth value
  - ° subsequentAuthReason= 7

- **REST:**
  - ° processingInformation.recurringOptions.firstRecurringPayment=true
  - ° processingInformation.authorizationOptions.initiator.credentialStoredOnFile = true
  - ° processingInformation.commerceIndicator = internet, MOTO or payer auth value
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason = 7

Recurring - Standing Order

A CIT made pursuant to an agreement between a cardholder and a merchant, whereby the cardholder authorizes the merchant to store and use the cardholder's account data **for a transaction that must occur at regularly scheduled intervals for a variable amount**, as specified in the agreement.

- **SCMP API:**
  - ° auth_first_recurring_payment= Y
  - ° subsequent_auth_first = Y
  - ° e_commerce_indicator = internet, MOTO or payer auth value
  - ° subsequent_auth_reason = 8

- **Simple Order API:**
  - ° firstRecurringPayment=Y
  - ° subsequentAuthFirst = true
  - ° ccAuthService_commerceIndicator = internet, MOTO or payer auth value
  - ° subsequentAuthReason= 8

- **REST:**
  - ° processingInformation.recurringOptions.firstRecurringPayment=true
  - ° processingInformation.authorizationOptions.initiator.credentialStoredOnFile = true
  - ° processingInformation.commerceIndicator = internet, MOTO or payer auth value
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason = 8

Mastercard MIT—Subsequent Transactions

Merchants must evaluate their businesses to determine whether their transactions correspond to the network-defined use cases discussed below. New values in the existing fields have been developed to identify the authorization reason according to the Industry Practice and Standing Instruction use cases.

Industry Practice MITs:

These MITs are known as 'Industry Practice' and are performed as a follow-up to an original cardholder-merchant interaction that could not be completed in a single transaction. Merchant-initiated transactions use stored credentials. The following transaction types are Mastercard industry practice transactions:

- ° Resubmission
- ° Delayed Charges
- ° Partial Shipment
- ° No Show

Resubmission

Previous attempt to obtain authorization for a transaction has been declined but the issuer's response does not prohibit the merchant from trying again later. Examples include insufficient funds/over credit limit response from the issuer or transit debt recovery.

- **SCMP API:**
  - ° e_commerce_indicator = Internet or MOTO
  - ° subsequent_auth = Y
  - ° subsequent_auth_reason = 1
  - ° subsequent_auth_stored_credential = Y
  - ° subsequent_auth_transaction_id = <auth_payment_network_transaction_id>

- **Simple Order API:**
  - ° ccAuthService_commerceIndicator = internet or MOTO
  - ° subsequentAuth = true

- ° subsequentAuthReason = 1
- ° subsequentAuthStoredCredential = true
- ° subsequentAuthTransactionID = < paymentNetworkTransactionID>

- **● REST:**
  - ° processingInformation.commerceIndicator = Internet or MOTO
  - ° processingInformation.authorizationOptions.initiator.type = merchant
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason= 1
  - ° processingInformation.authorizationOptions.initiator.storedCredentialUsed = true
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionID = < processorInformation.networkTransactionId >

Delayed Charge

An additional account charge after initial services have been rendered and payment has been processed. For example, a hotel mini bar charge after the cardholder has checked out of the hotel.

- **● SCMP API:**
  - ° e_commerce_indicator = Internet or MOTO
  - ° subsequent_auth = Y
  - ° subsequent_auth_reason = 2
  - ° subsequent_auth_stored_credential = Y
  - ° subsequent_auth_transaction_id = <auth_payment_network_transaction_id>

- **● Simple Order API:**
  - ° ccAuthService_commerceIndicator = internet or MOTO
  - ° subsequentAuth = true
  - ° subsequentAuthReason = 2
  - ° subsequentAuthStoredCredential = true
  - ° subsequentAuthTransactionID = < paymentNetworkTransactionID>

- **● REST:**
  - ° processingInformation.commerceIndicator = Internet or MOTO
  - ° processingInformation.authorizationOptions.initiator.type = merchant
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason= 2
  - ° processingInformation.authorizationOptions.initiator.storedCredentialUsed = true
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionID = < processorInformation.networkTransactionId >

Partial Shipment Transaction

Occurs when an agreed quantity of goods ordered through e-commerce are not all available for shipment at the time of purchase. Each shipment is a separate transaction. For example, consumer ordered goods that ship at different times.

- **● SCMP API:**
  - ° e_commerce_indicator = Internet or MOTO
  - ° subsequent_auth = Y
  - ° subsequent_auth_reason = 6

- ° subsequent_auth_stored_credential = Y
- ° subsequent_auth_transaction_id = <auth_payment_network_transaction_id>

- **Simple Order API:**
  - ° ccAuthService_commerceIndicator = internet or MOTO
  - ° subsequentAuth = true
  - ° subsequentAuthReason = 6
  - ° subsequentAuthStoredCredential = true
  - ° subsequentAuthTransactionID = < paymentNetworkTransactionID>

- **REST:**
  - ° processingInformation.commerceIndicator = Internet or MOTO
  - ° processingInformation.authorizationOptions.initiator.type = merchant
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason= 6
  - ° processingInformation.authorizationOptions.initiator.storedCredentialUsed = true
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionID = < processorInformation.networkTransactionId >

No-Show Transaction

A penalty charged according to the merchant's cancellation policy. For example, a cardholder's cancellation of a reservation without providing proper advance notice to the merchant.

- **SCMP API:**
  - ° e_commerce_indicator = Internet or MOTO
  - ° subsequent_auth = Y
  - ° subsequent_auth_reason = 4
  - ° subsequent_auth_stored_credential = Y
  - ° subsequent_auth_transaction_id = <auth_payment_network_transaction_id>

- **Simple Order API:**
  - ° ccAuthService_commerceIndicator = internet or MOTO
  - ° subsequentAuth = true
  - ° subsequentAuthReason = 4
  - ° subsequentAuthStoredCredential = true
  - ° subsequentAuthTransactionID = < paymentNetworkTransactionID>

- **REST:**
  - ° processingInformation.commerceIndicator = Internet or MOTO
  - ° processingInformation.authorizationOptions.initiator.type = merchant
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason= 4
  - ° processingInformation.authorizationOptions.initiator.storedCredentialUsed = true
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionID = < processorInformation.networkTransactionId >

Standing-Instruction MITs

Standing-Instruction MITs are performed in order to follow agreed-upon instructions from the cardholder for the provision of goods or services. The following transaction types are standing instruction transactions:

- ° Installment
- ° Subscription
- ° Standing Order
- ° Unscheduled Credential on File (COF)

Installment Payment

A transaction in a series of transactions made pursuant to an agreement between a cardholder and a merchant, whereby the cardholder authorizes the merchant to store the cardholder's account data for use by the merchant to initiate one or more future transactions for a **known amount over a specified duration based on a single purchase.** For example**,** a cardholder purchases a television for $1,000 USD, pays in four equal installments of $250 USD (first transaction is CIT, remaining three transactions are MIT.)

- **• SCMP API:**
  - ° e_commerce_indicator = install
  - ° subsequent_auth = Y
  - ° subsequent_auth_stored_credential = Y
  - ° subsequent_auth_transaction_id = < auth_payment_network_transaction_id >
  - ° subsequent_auth_reason = 9

- **• Simple Order API:**
  - ° ccAuthService_commerceIndicator = install
  - ° subsequentAuth = true
  - ° subsequentAuthStoredCredential = true
  - ° subsequentAuthTransactionID = < paymentNetworkTransactionID>
  - ° subsequentAuthReason = 9

- **• REST:**
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionID = < processorInformation.networkTransactionId >
  - ° processingInformation.authorizationOptions.initiator.storedCredentialUsed = true
  - ° processingInformation.authorizationOptions.initiator.type = merchant
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.reason = 9
  - ° processingInformation.commerceIndicator = install

Recurring Payment – Subscription

A transaction made pursuant to an agreement between a cardholder and a merchant, whereby the cardholder authorizes the merchant to store and use the cardholder's account data for a transaction that must occur at **regularly scheduled intervals** for a **fixed amount**, as specified in the agreement. The subscription agreement may include an allowance for introductory pricing or for price changes to occur

from time to time. For example, a monthly subscription or fixed monthly service payment (like a newspaper.)

- **SCMP API:**
  - ° e_commerce_indicator = recurring
  - ° subsequent_auth = Y
  - ° subsequent_auth_stored_credential = Y
  - ° subsequent_auth_transaction_id = < auth_payment_network_transaction_id >
  - ° subsequent_auth_reason = 7

- **Simple Order API:**
  - ° ccAuthService_commerceIndicator = recurring
  - ° subsequentAuth = true
  - ° subsequentAuthStoredCredential = true
  - ° subsequentAuthTransactionID = < paymentNetworkTransactionID>
  - ° subsequentAuthReason = 7

- **REST:**
  - ° processingInformation.commerceIndicator = recurring
  - ° processingInformation.authorizationOptions.initiator.type = merchant
  - ° processingInformation.authorizationOptions.initiator.storedCredentialUsed = true
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionID = < processorInformation.networkTransactionId >
  - ° processinginformation.initiator.merchantInitiatedTransaction.reason = 7

Recurring Payment –Standing Order
A transaction made pursuant to an agreement between a cardholder and a merchant, whereby the cardholder authorizes the merchant to store and use the cardholder's account data for a transaction that must occur at **regularly scheduled intervals** for a **variable amount**, as specified in the agreement. For example, a monthly utility payment.

- **SCMP API:**
  - ° e_commerce_indicator = recurring
  - ° subsequent_auth = Y
  - ° subsequent_auth_stored_credential = Y
  - ° subsequent_auth_transaction_id = < auth_payment_network_transaction_id >
  - ° subsequent_auth_reason = 8

- **Simple Order API:**
  - ° ccAuthService_commerceIndicator = recurring
  - ° subsequentAuth = true
  - ° subsequentAuthStoredCredential = true
  - ° subsequentAuthTransactionID = < paymentNetworkTransactionID>

- ° subsequentAuthReason = 8

- **REST:**
  - ° processingInformation.commerceIndicator = recurring
  - ° processingInformation.authorizationOptions.initiator.type = merchant
  - ° processingInformation.authorizationOptions.initiator.storedCredentialUsed = true
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionID = < processorInformation.networkTransactionId >
  - ° processinginformation.initiator.merchantInitiatedTransaction.reason = 8

Unscheduled Credential on File (COF)
A transaction made pursuant to an agreement between a cardholder and a merchant, whereby the cardholder authorizes the merchant to store and use the cardholder's account data to initiate one or more future transactions. The transaction can be for a fixed or variable amount but does not occur at regular intervals on a scheduled date. For example, the consumer agrees to allow a tollway merchant to initiate transactions when the balance in the consumer account with the merchant falls below an established amount (auto top-up.)

- **SCMP API:**
  - ° e_commerce_indicator= Internet or MOTO
  - ° subsequent_auth=Y
  - ° subsequent_auth_stored_credential = Y
  - ° subsequent_auth_transaction_id = < auth_payment_network_transaction_id >

- **Simple Order API:**
  - ° e_commerce_indicator = Internet or MOTO
  - ° subsequentAuth = true
  - ° subsequentAuthStoredCredential = True
  - ° subsequentAuthTransactionID = < paymentNetworkTransactionID>

- **REST:**
  - ° processingInformation.commerceIndicator = Internet or MOTO
  - ° processingInformation.authorizationOptions.initiator.type = merchant
  - ° processingInformation.authorizationOptions.initiator.storedCredentialUsed = true
  - ° processingInformation.authorizationOptions.initiator.merchantInitiatedTransaction.previousTransactionID = < processorInformation.networkTransactionId >

Mastercard Trace ID
The Trace ID of the originating CIT transaction must be included for all MIT scenarios:
- ° Resubmission
- ° Delayed Charges
- ° Partial Shipment

- ° No Show
- ° Installment
- ° Subscription
- ° Standing Order
- ° Unscheduled COF

Possible Values for Subsequent Authorization reason:

- ° 1 = Resubmission
- ° 2 = Delayed Charge
- ° 4 = No show
- ° 6 = Partial Shipment
- ° 7 = Subscription
- ° 8 = Standing Order
- ° 9 = Installment

Supported Gateways:

- ° VPC
- ° FDC Compass