Enrollment Widget

This document is intended for Mastercard partners willing to integrate their mobile applications or website solutions with Mastercard Reward System using Enrollment Website/Widget tool provided by Verestro.

The document is designed to cover the following fundamentals:

- High level overview of Enrollment Widget capabilities and architecture,
- Integration description,
- Data flow diagrams,
- Illustrative User flows and Screens.

Abbreviations:

AES – Advanced Encryption Standard ,

API – Application Programming Interface,

MRS - The Mastercard Rewards System - the platform from Mastercard dedicated to deliver loyalty solutions for customers,

PCI DSS - Payment Card Industry Data Security Standard.

Overview

Online Enrollment Capability (Enrollment Website/Widget) is a tool that can be integrated with Merchants, Issuers or other Clients and allows secure registration of cardholder PAN (and PII data, if applicable) into the MRS system. In addition to card enrollment (and PII, if applicable) into MRS, it also allows to perform certain card management activities (optional), including unenrollment and card replacements.

The Enrollment Capability (Enrollment Website/Widget) designed by Verestro supports all key web-browsers (see further in the document) and has mobile responsive design, as well as can be embedded into mobile applications (WebView). Also, it can be customized & localized per specific Client/Country (with unique URL) and enabled upon specific request from Mastercard.

Verestro is using MRS API v.2.0 to facilitate operation of Enrollment Capability (Enrollment Website/Widget). The Enrollment Capability will be hosted in Poland (EU) in Verestro PCI DSS certified environment (hosting center).

Limited PII data elements, stored in Verestro database are encrypted using AES 256 encryption standard. PAN is not stored in Verestro database at any time.

High Level Description of Flow and Requirements

Enrollment

Initialization Process

In order to start using Verestro solution, each Client is required to be setup in the Verestro back-end system. The Client setup process includes assignment of unique Client ID, Program ID, and Security Key exchange process.

In order to initialize Enrollment Website/Widget capability, Client will be required to trigger a dedicated URL provided by Verestro with parameters included into signed request described in 3.2 “Initialization process” (HTTP POST action /company/non_auth_initialize request with JWT token (RFC 7519) which contains valid payload data). For any new registration, payload should contain valid actionCode parameter that is equal to “N” value for new customer/card enrollments.

Please note – Client is required to authenticate/verify the customer and submit valid Customer ID in the initialization request to Verestro. If the Client does not provide Customer ID during the initialization of enrollment, Verestro will generate a Customer ID on behalf of the Client. In such case, Verestro will provide back to the Client the assigned Customer ID value and will display assigned value to the customer. The Customer ID enables further card management purposes such as e.g. opt-out) but is not a default option and will require additional security measures on Verestro side (including Re-captcha and/or 3DS process).

Upon successful validation on Verestro side, the Website/Widget is displayed, where user can enter the following data:

- PAN (Required),
- First Name (Optional),
- Last Name (Optional),
- Email (Optional),
- Terms & Condition consent (Optional, used in majority of cases),

- Privacy Notice consents (Optional, used in majority of cases).

The PAN during customer's input in Website/Widget is validated using Luhn algorithm in real-time (in the browser). If successful Luhn check is passed in the browser, Verestro will encrypt the PAN using MC public key (see full process in section Security 3.3) and will pass the encrypted card information into MRS. After successful MRS enrollment, MRS will supply back to Verestro successful enrollment notification with Account Ref ID or RANAC (unique ID assigned by MRS per card) for further card management activities.

In addition, Verestro will be required to immediately feedback the enrollment result with assigned values to the Client (Customer ID, Account Ref ID or RANAC, additional values if required). Partner can use the one of initialization parameters (ranac_url) to send a specific endpoint to which Verestro will send RANAC after successful enrollment.

3DS authentication *(optional)*

Optionally, Verestro allows to trigger 3DS 2.0 authentication after submission of the registration data. If the card authentication is successful, the card enroll is performed into MRS.

This case must enable the decryption of the card on the API side.

Un-Enrollment User Flow

In order to initialize the Enrollment Website/Widget to execute un-enrollment, Client will be required to trigger dedicated URL provided by Verestro with required parameters included into signed request described in 3.2 "Initialization process". In this case actionCode parameter should contain "C" value and Customer ID value is always required.

Verestro system will perform a search of Account Ref ID or RANAC assigned to Customer ID in Verestro database (decrypt stored values) and will trigger updateCustomerAccount MRS API with "CANCELLED" status. Upon successful un-enrollment in MRS, Verestro will immediately feedback the result of un-enrollment to the Client.

After X days from the status change to Canceled, the record with the any associated PII data (including Customer ID, Account Ref ID or RANAC, others) will be completely removed (deleted) from Verestro database. Please note – if there are multiple cards under single Customer ID, Verestro will be required to search Account Ref ID or RANAC having last 4 digits of card to perform card un-enrollment under associated Customer ID (only Account Ref ID or RANAC will be purged upon cancelation of card).

X – it is parameter configurable per Client/Program (e.g. 30 days).

Replacement User Flow

In order to initialize the Enrollment Website/Widget to execute replacement, Client will be required to trigger dedicated URL provided by Verestro with required parameters included into signed request described in 3.2 “Initialization process”. In this case actionCode parameter should contain “R” value and Customer ID is always required.Verestro system finds the cards assigned to this Customer ID in Verestro database and display the cards list in the following format:

- 1234 XXXX XXXX 1234

User can select the card he wants to replace and enters a new PAN.

Upon selection of card to replace, Verestro will propose to enter a new PAN. Verestro will capture & validate a new card number (in browser) and will trigger the new card enrollment into MRS (2.1 Enrollment). Upon successful enrollment of the new card, the cancelation of the old card will be triggered by Verestro into MRS (sequence will be followed). If by any reason, the card enrollment of the new card is not successful, Verestro will not delete the old card and will inform Client about the unsuccessful replacement attempt.

Verestro will immediately feedback the results of replacement including Customer ID, new Account Ref ID or RANAC (additional data if any) to the Client and confirm the successful replacement of old card.

After X days from the replacement, the record with the any associated PII data (including Customer ID, Account Ref ID or RANAC, others) will be completely removed (deleted) from Verestro database. X – it is parameter configurable per Client/Program (e.g. 30 days).

Widget Customization & Localization

Some parts of Enrollment Capability (Enrollment Website/Widget) can be customized per each integrating partner:

- Logotype.

  Supported format is: PNG. Supported proportion is 21:9 with transparent background. Minimum height is 100px.

- Background image.

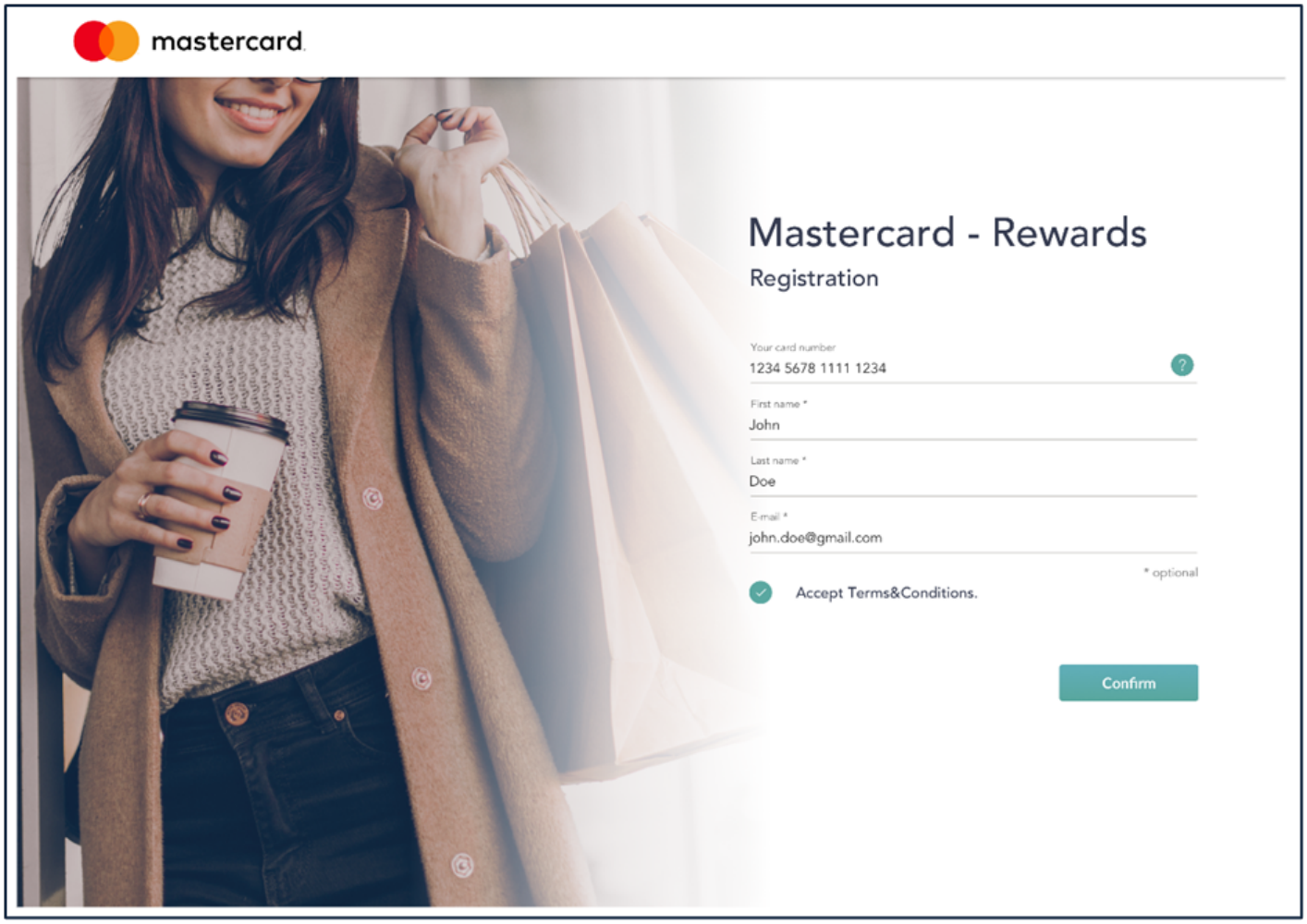
  Supported format is: PNG Supported resolution is: Full HD (1920px x 1080px)

- Text/Translations.

  The client will receive a translation file in JSON format, example below.

```
{
```

```
  "register":{
    "header":"Mastercard - Rewards",
    "tittle":"Registration",
    "accept_terms":"Accept Terms&Conditions",
    "userData":{
      "card_number":"Your card number",
      "first_name":"First name",
      "last_name":"Last name",
      "email":"E-mail"
    },
    "optional":"optional",
    "confirm":"Confirm"
  },
  "error":{
    "title":"Something went wrong...",
    "info_first_part":"Your card does not belong to the program. Read more about how to join the program.",
    "info_second_part":"Incident identifier: "
  },
  "success":{
    "title":"Success!",
    "info":"Your card has been attached to the program."
  },
  "read_more":"Read more"
}
```



Browsers supported

Enrollment Website tool works on modern browsers including the most recent versions of Chrome, Firefox, Safari, and Microsoft Edge. Older legacy versions may still work, but for an optimal experience, please use the most up to date browser versions.

Technical description of the setup

Before configuring the Client on the Verestro side, Mastercard should provide a complete set of data for specified for the Partner (Client) and Program:

- Partner Name,
- Partner Language/Country (Translation File),
- Terms & Conditions or Privacy Notice (URL or Content),
- Type of Bank Customer Number,
- Member ICA,
- Bank Product Code,
- Program Identifier,
- List of BINs.

Initialization Parameters

In order to initialize the Enrollment Website/Widget, Client is required to make HTTP POST action /company/non_auth_initialize request to Enrollment Website (API) with JWT token (RFC 7519 algo: RS-256) which contains valid payload data signed with secret key.

Service URLs:

Test: https://rpm.upaidtest.pl

Prod: https://rpm.upaid.pl

Token signing

Client should generate key pair using commands:

openssl genrsa -out private-key.pem 2048

openssl rsa -in private-key.pem -pubout -out public-key.pem

Client provides public key for Verestro. Client signs JWT token with private key. Verestro will verify token signature by provided public key.

Headers:

The endpoint expect the following headers:

Name	Value
Accept	application/json
Content-Type	application/json

Client should execute HTTP POST /company/non_auth_initialize method with initializeWebsiteToken parameter in the JSON body.

JSON Body example:

{

"*initializeWebsiteToken*": "*eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJtaWxlc2FuZG1vcmUiLCJzdWIiOiJ0ZXN0YWhkZmprc2RoZmciLCJwcm9ncmFtSWQiOiIxMDc2NSIsImFjdGlvbkNvZGUiOiJOIiwiY29uZmlybWF0aW9uVXJsIjoiaHR0cDpcL1wvdGVzdHVwZGFpZC5wbCIsImxhbmd1YWdlIjoicGwiLCJleHAiOjE2MTI5NTg3NTJ9.FV13fKPst5DLhCOz4VLOoeSUjKGtOxMlceJtDtXE_8Fa498fnP3DWqK763AQNF0U32UDeq10X6ctUYKxe2-xwTMFdBe8PU2xmz-khRFfV0l0fz9J3xFGjR59PTBdmYzLWJ5AmU5EDg4SpWCT4Oaobq2eBYJ_WGO7MKDx_7okFa7Z_H1OjM

OAone3OSJWIY84J9rmeqt3GvD5r7CwewReExGI15MBy5VfUqh5_543b5gNjJgTeYBGha46DdtFChk7mJNfjNQwGMcqJZsjlCxoqRbWC9Jcz0T0eLfZBGLMeSszSwfXgKqM0NelFhVUSU99kFLvl8MSBcz1j6yhG6VziQ"

}

HTTP Request parameters:

Name	Required	Type	Description
initializeWebsiteToken	Yes	string	JWT token signed by Client private key

Token payload data:

Name		Required	Type	Description.
	iss	Yes	string	Partner ID (Client ID) – unique value assigned by Verestro per Client that is required to display the proper website/experience.
	sub	Yes	string	External Customer ID - Unique Customer Identifier (unique Customer ID) assigned by the Client (e.g. Customer ID, Parent ID). Please note – Client is required to authenticate/verify the customer and submit Customer ID in the initialization request. If the Client does not generate Customer ID during the initialization of enrollment, but it is not default option and will require additional security measures on Verestro side (including Re-captcha and/or 3DS process).

	actionCode	Yes	string	Unique action code for optional functionally: · N – New registration, · R – Replacement, · C – Opt-out.
	programId	Yes	string	Program ID – unique value assigned by Mastercard for specific Program.
	iat	Yes	numeric	The "iat" (issued at) claim identifies the time at which the JWT was issued. This claim can be used to determine the age of the JWT. Its value MUST be a number containing a numeric date value. Eg. 1516239022.
	exp	Yes	numeric	The "exp" (expiration time) claim identifies the expiration time on after which the JWT MUST NOT be accepted for processing. Please set token lifetime to 10 min.
	language	No	string	Partner preferred language code. If field is not specified then is return default language for given Partner. Eg. en_US (IETF language tag ISO 3316 & ISO 639).
	confirmationUrl	No	string	The URL that will allow sending to partner backend status and parameters like RANAC (REFID from MRS API) parameter or other error codes allowing full tracking.

Token payload example:

```
{

  "sub": "CustomerID",
  "actionCode": "N",
  "programId": "ProgramID",
  "iss": "ClientID",
  "iat": 1516239022,
  "exp": 1516269622,
  "confirmationUrl": https://confirmation-sys.com/id/123,
  "language": "en_US"

}
```


Response

This request will always respond with HTTP 302 code which is redirect status response. It’s because at first you hit a security layer application that translate tokens and grants access to main application.

The URL returned in HTML tag should be use to open the widget.

Security Details

General Security Measures

- Enrollment Website/Widget will be hosted in Poland (EU) in Verestro PCI PSS certified environment (hosting center).
- Vulnerability scans are performed quarterly on Verestro systems and external, certified PCI DSS auditor performs annual audit (latest PCI DSS certification has been successfully passed in October 2019).
- All systems are backed-up.
- Security policies are available anytime for Mastercard and its partners.
- All PII data elements, stored in Verestro database, are encrypted using AES 256 encryption standard. All parameters provided by the client at point of enrollment or card management (initialization) must be signed. PAN is not stored in Verestro database at any time - unique MRS card IDs will be assigned for any further activities with account.

PAN encryption process

The PAN is not stored in Verestro database at any time and the following PAN enrollment process is followed by Verestro:

1. CH inputs PAN on the browser (enrollment page/widget hosted by Verestro), where Verestro will perform auto-check of Luhn to ensure correctness of card number input (still inside the browser - javascript). If supplied PAN number doesn’t pass the Luhn check, a notification will occur to cardholder to correct the PAN number.
2. If successful Luhn check is passed in the browser, Verestro encrypts the PAN using MC public key (MTF – refid: 119009, Production – refid: 122744) and will pass the encrypted card information into https to backend API (Verestro) to further trigger enrollment of card data into MRS. The PAN is the field being encrypted. Verestro will take the encrypted data from the browser and will put it unchanged (encrypted) into the field along with the indicator that it is an encrypted value.

Example:

An unencrypted PAN would be sent like this:

<cus:bank_account_number>

5xxxxxxxxxxxxxx</cus:bank_account_numbe>

An encrypted PAN would be sent like this:

<cus:bankAccountNumber encrypted="true">

{"key":"<received from browser>",

"data":"<received from browser>"}</cus:bankAccountNumber>

The values for “key” and “data” would be generated in the browser and sent to Verestro to relay on to us.

3. Verestro backend will send this encrypted value (encrypted in browser) on to MRS within an enroll MRS API call.
4. MRS will decrypt using MRS private key to check eligibility (incuding BIN-ranges) and upon successful check enroll PAN into MRS. After successful enrollment into MRS, MRS will supply back to Verestro successful enrollment notification with Account Ref ID or RANAC (unique per card) assigned for account (for further activities, if applicable).
5. Other PII data (e.g. Customer IDs) will be stored by Verestro under AES 256 encryption. uPaid will also store Account Ref ID but at no time full PAN data.

Data storage

Any PII data (e.g. Customer ID), will be stored by Verestro only under AES 256 encryption that will allow to decrypt the PII data when there is a need to fulfill certain business requirements. In addition, Verestro will store Account Ref ID/RANAC under AES 256 (unique card ID assigned by MRS) but at no time full PAN data.

Communication from website

In general, all communication to and from the website we can list in a few categories below.

- Initialize.
  - Partner not allowed - Signature for the token is invalid, or the partner has no access to that functionality.
  - Invalid parameters - Some of the required parameters are invalid or required ones are not present.
- Ending confirmation – By JS events or HTTPS backend to backend request.
  - Success - Enrollment, replacement, or removal was performed successfully.
  - Error - There was some problem during enrollment, replacement, or removal.

The widget can be used in a couple of ways, eg.

- Web redirect (full page),
- Web iframe,
- Mobile WebView.

Widget supports two ways of feedback to partners.

- JS Event for WebView,
- HTTPS request for all above, optionally.

Name	Required	Type	Description.
status	Yes	string	Main status, all possibilities listed above.
status_code	No	string	MRS error code, present only when the source of error is MRS API.
parentId	No	string	External Customer ID - Unique Customer Identifier.
ranac	No	string	Unique ID assigned by MRS per card.
designCode	No	string	Code representing card visual design. Eg. Gold, Blue.
bin	No	string	First 6 digits of PAN aka bin.
last4	No	string	Last 4 digits of PAN.

All the supported processes can have main statuses:

- Partner not allowed,
  - ACCESS_DENIED,
  - TOKEN_EXPIRED,

- Invalid parameters,
  - INVALID_PARAMETERS,
- Success,
  - SUCCESS_ENROLLMENT,
  - SUCCESS_REPLACEMENT,
  - SUCCESS_REMOVAL,
- Error,
  - ERROR_ENROLLMENT,
  - ERROR_REPLACEMENT,
  - ERROR_REMOVAL.

Application support optional confirmation using secure backend to backend system HTTPS requests. This is optional, a partner can rely only on data from JavaScript event, or use both, depends on preferences. This request is sent, only when parameter *confirmationUrl* is present in the initialization process.

For each interaction with MRS (enrollment, replacement, or delete) system will send a request with the overall process status with all necessary parameters described below.

All potential statuses from MRS, that can be in the STATUS_CODE field, listed below:

Code	Description.
0	Success .
-1	Invalid parameters (e.g. when invalid Bank Product Code is sent, but it should not happen as Verestro is hard coded).
-5	Invalid Account was not found within MRS (e.g. if Veresto will send invalid RANAC for replacement, but it should not happen as Verestro has it mapped on their side).
-10	Invalid Account Status (should not happen as Verestro knows the status of account).
-32	Invalid Program ID (e.g. when Veresto will send the invalid Program ID to enroll the card, but should not happen as Verestro is hard coded to single ID).
-44	Invalid Bank Account Number (e.g. when PAN doesn’t pass Luhn validation, impossible as Verestro checks in browser during input).
-45	Bank Account Number Already Enrolled (e.g. when PAN is already enrolled under different Parent ID).
-46	Invalid Bank Product Code (e.g. when BPC is not valid, should not happen as Verestro is hard coded).

-47	Invalid Account Status Code (e.g. when non-standard code input, but Verestro knows all our status codes for enrollment and cancelations).
-48	Invalid Program Identifier (when invalid Program Identifier input, but Verestro knows and coded to single Program Identifier so should not happen).
-50	Bank Customer Number Already Enrolled (when Parent ID already enrolled, should not be a issue, as we have 1-1 customer account approach + Verestro keeps track of Parent IDs).
-52	Account not qualified for the Program (when card doesn't fall under DKB co-brand range, but should not happen as initially Verestro checks it).
-85	Bank account number does not exist (when RANAC is not enrolled, but should not happen as Verestro has a track of RANACs with Parent IDs).
-1001	Server Side Error. Internal Error (Connection to MRS error).

Example confirmation request:

HTTPS request (POST) to confirmationUrl

Body (JSON):

```
{
  "status":"SUCCESS_ENROLLMENT",
  "statusCode":"-50",
  "parentId":"5d7703f9-7df2-4323-9745-9474f3182aef",
  "ranac":"03643782610476418733",
  "designCode":"Gold",
  "bin":"523400",
  "last4":"0123"
}
```

Example JS event integration for communication widget à native mobile application.

Event name: "enrollmentEnded" for all type of actions.

Event data structure:

```
{
  status: "SUCCESS_ENROLLMENT",
  statusCode: "-50",
```

```
  parentId: "5d7703f9-7df2-4323-9745-9474f3182aef",
  ranac: "03643782610476418733",
  designCode: "Gold",
  bin: "523400",
  last4: "0123"
}
```

Revision #4
Created 7 July 2022 19:18:55 by Barbara Tudruj
Updated 26 August 2022 09:01:55 by Jagoda Mazurek