Overview

Verestro Card Management System is called ANTACA. The platform provides solutions for creating and managing users' accounts (called "balances"), processing eKYC (user authentication process) and issuing payment cards generated for them.

CMS Antaca provides dedicated services for:

- end-user mobile applications,
- server-to-server connections helpful in integration with existing customer databases,
- administrative panel, necessary from the point of view of financial institutions in the process of issuing cards and managing their clients' funds.

CMS Antaca supports all necessary use cases for various digital and plastic card issuing. It supports integration with multiple issuing processors and can be connected with the one chosen by Verestro partner.

Introduction to Card Issuing process

With the CMS Antaca you can offer your customers three types of cards:

1. Virtual card - Digital card without any physical components.
2. Physical card – The traditional plastic payment card.

To be able to issue a card for a user, 4 requirements must be met:

1. You have to integrated with Verestro platform using JWE token (described below) or other integration methods (API, SDK, White Label).
2. User must exist in Verestro database called DataCore. Make sure you register user via User Lifecycle API & SDK.
3. User must be strongly verified according to KYC. You can use Verestro KYC (see below) or own KYC process.
4. The user must have a User Balance under which the card will be generated.

After those 4 steps you can issue a card for the User.

Below we describe this process step by step:

- Step 1. Configuration & JWE Security,
- Step 2. User Lifecycle API & SDK,

- Step 3. User registration & KYC,
- Step 4. Create User Balance (account),
- Step 5. Card issuing.

Terminology

Name	Description
Customer	Institution which is using Verestro products. This institution decides which SDK should be used and how transaction should be processed. Basicly Customer can be called Verestro client.
User	User which is using Payment Hub Application. It is root of entity tree. User is identified in Wallet Server by some unique identifier which is provided after registration. User can have access to his data and operations based on session. User’s session is created after device pairing is performed. When session expires then user authentication have to be performed. Session is valid 10 minutes, however it is configurable parameter.
Card	Card belongs to the user. User can have many cards. Card is identified via internal id given after storing card on Wallet Server. Whole PAN is stored on Wallet Server which has PCI DSS certificate.
Device	Device belongs to user. When user starts using application after installation then device pairing is performed. After pairing device with some unique id, unique device installation id is generated and this installation is assigned to user. It is possible to have one active installation on specific device for specific user.
Session Token	Token which defines User. It is an authorization way of the User. This entity is created after paring device and this is needed to perform any actions in the application. When session is expired then user authentication needs to be performed. Session is valid 10 minute s, however it is configurable parameter.
Sender	Verestro Wallet user which triggers transaction to the Receiver (check User description).

Receiver	Receiver can be identified in Wallet Server (Internal) or may be an entity that does not exist in Wallet Server (External). ◦ Internal – this type of Receiver has his own unique identifier just like sender. It can also act as a Sender in the transaction process, ◦ External – this type of Receiver does not exist in Wallet Server. Transfers that are made to this type of Receiver require the entering of his card data by Sender.
Mid	Merchant identifier. This entity is representing Merchant in Acquirer’s system. Customer have to provide the mid information to enable mid configuration in the Verestro system. Required to process 3DS authentication via Verestro System.
Acquirer	External institution responsible for processing transaction and 3ds requests ordered by the Verestro Payment Hub App. Acquirer connects with banks / card issuers and returns information whether the ordered action on a given card is possible.
PAN	(Primary Account Number) It is 14-19 (usually 16) digits number which is a unique identifier of the payment card issued to the customer's account.
Wallet Server	Provides the backend services to support Mobile Payment Application via Verestro Wallet SDK and is responsible for managing users, devices, cards , device tokens, storing transactions history and communication with Acquirers.
PCI DSS	PCI DSS (Payment Card Industry Data Security Standard) is a security standard used in environments where the data of payment cardholders is processed. The standard covers meticulous data processing control and protection of users against violations.
IBAN	IBAN (International Bank Account Number) is an international standard for bank account numbering that allows you to transfer funds to foreign accounts and to receive transfers from foreign entities to domestic bank accounts. One of the assumptions of the IBAN standard is to simplify the system of cross-border transfers.
QR	A QR code (quick response code) is a two-dimensional barcode. Check here for more details.

Configuration & JWE Security

To start the implementation, it is necessary to configure the payment processor. If we are using issuing processors already integrated with Verestro the process is simple and after quick information gathering (name of partner, BIN range, currency, remoteURL) a new card program can be setup for our partner.

You can communicate with the CMS Antaca API in three different dedicated channels:

1. Mobile Application - Methods strarting with /Customers : designed for the mobile applications that use a session token sent in the header of each request. More about the possibilities of generating these tokens in the section White Label Application Overview.
2. Server-to-server - methods starting with /Secure : this communication channel is protected by the x509 certificate. To start an implementation based on this communication channel, it is necessary to generate your own CSR and send it to Verestro. Verestro will sign it and return a valid certification in a response.
3. Administrator and Customer Service (rarely used by partners) - methods starting with /admin : designed for the administration panel provided by Verestro.

Additional data encryption & integration

Some requests and responses contain sensitive data, to additionally secure the connection we require JSON Web Encryption (JWE).

	normal	encrypted

Example of request with sensitive data.	`{`
	`"cardNo" : "5555444455554444"`
	`}`
		`{`
		`"payload" : "very long JWE token"`
		`}`

Example ofresponse with sensitive data.	`{`
	`"id" : 1125,`
	`"type" : "1125" ,`
	`"cvv" : "123" ,`
	`"cardNo" : "5555444455554444" ,`
	`"exp" : "2026-01-31"`
	`}`
		`{`
		`"payload" : "very long JWE token"`
		`}`

JWE configuration

To setup connection we need from you enc and alg from JWE parameters. Acceptable values are:

- Algorithm used by Verestro to encipher content of message (enc) - A256GCM,
- Algorithm used by Verestro to encipher encryption key (alg) - RSA-OAEP-256,
- Algorithm needed from you to encipher content of message (enc) - A256GCM,
- Allowed algorithms for key encryption (alg) - RSA-OAEP-256 or RSA-OAEP.

Recommended JWE libraries for various programming languages:

- PHP,
- JAVA.

Request

To process encrypted message you need to perform a few additional steps on top of standard message processing:

- Add headlines:
  - **Public-Key** through which you can transfer to us your public key encoded b64 (more details below),
  - **Encrypted-Request** headline confirming message encryption in both directions or **Encrypt-Response** when you need to get the encrypted response only; value true or false,
- Download Verestro Public Key - see in technical API specs on which endpoint,
- Use Verestro Public Key to create JWE and transfer data table in payload,
- Use token (string) received in Verestro response in point 3 below key encryption key in payload.

Additional information:

- for GET methods avoid point 2, 3, 4 above (headlines mentioned in point 1 are still necessary),
- for empty POST methods (without "body") use same rules as for GET message.

Response

After sending to CMS Antaca encrypted request you will receive from us encrypted message:

1. Decipher token, which can be found in response below payload key (use your private key to perform this action),
2. After decipher action you can see response in unencrypted form.

Additional information:

- Response are encrypted only in case of success - HTTP 20X,
- The only exception from the above mentioned rule is code 204 No content,
- In case of errors (i.e. validation errors) you will receive unencrypted response,
  - ENCRYPTION_REQUIRED,
  - INVALID_PUBLIC_KEY,
  - INVALID_PAYLOAD,
  - CANT_DECRYPT_PAYLOAD.

Example request:

Correct request	Sent request (incorrect)	Received by CMS Antaca (after decipher action with private key)
`{"card_no" : 1337}`	{"payload" : "eyJhbGciOiJSU0EtT0FFUC0yNTYiLCJlbmMiOiJBMjU2R0NNIn0.rdUrW12XCZQgLFDJ-2zAHWYYnaAanctceE1-Y6yJUplX0B2dLu-bvYOEJ83KxxUs-ZjA41R4PmAVilx1cTF4pv-7CZR0_ki85XRATBYF2-MvZdcC81fHy2QPU_ZsAEWAW00a1wKJmuEsgPB2m1aLZ7oK4fC1hciep4PyAtuWQRYHjhNb-UDT41_gDKTbnSGTwheL7S0mAJ_HsKfnZFHYUrM77UcxQGZKnH7Mzqvndf9THiMo0-3MWliYFDAm1bqN2_KTIoBNCprYjFnyIXPCjib73bjWX_P2ip5Ul84cngbQmFVzc7o91JrpJvYou1INS7zL4XKLFcADN4nZ_9ePWsm5_kX5SOMyUyEhOC9gusrLNAJ0MHaIFHni8WqnMAWM3_MC4OQDYetKax5bnHK6x42_5eFaf6ZmzmioKny5aGm-4Vo8TEu691FmPxglhyenWlMhvBvf6ZeVsy58Ofr0mi3TXjwYbAyas7m6sncxZu1FhEJ4da6gtNjmjuKdikOOntu8V71QQ07nczNqfGlUv0RcUc9uKJq5je4b9BEbK9WuQcroxmALqC4HTt1xhICHrVUA0d_t3fglhS2n7wNaKKCFq70ZWIrpdTaBd35kdVQOEjZgCavSjbZOzgOzcEqS6P2Blm7bZ7ZZBmnfk8y8M4m0xWoQNTmLC6nqz9bSbME.UEryKNClDxQZpyWu.6Lw_5CcZ9HiVxHfi_XTAFw.pYbQ6tdmQYe1kiPonm1GhA"}	`{"card_no" : 1337}`

Example response:

Correct response	Sent response by CMS Antaca	Received by you

`{"card_no" : 1338}`	{"payload" : "eyJhbGciOiJSU0EtT0FFUC0yNTYiLCJlbmMiOiJBMjU2R0NNIn0.iPmvEKtMAMrrEiR89vlwsL77ZfqxXrcMiy-bx3z6_7HAo__aQzBpMVDtLyj3kTHYWxen8bhPuVyebXyaIHL20sekFzcIFFzvaGoyQYU6zOK8tPv81tgixQe8SDnEr5v9VWBfiHxtPvqlpQIig2is5ynBkyqjdpQWEagR3MpqpATGl7f-omG82Jq0OwZByWI8I6P89hczwgK37F-MUnQDxcRUM3RagbHKNeIcfmPdJpNeqFZHe45y4wUkTWN0uzW72qydkN_4uM9fy0nrUpgsJNbtJGAVIUVmDz4pIZkiI1zyGbfZX-PT7Wh9UNM06gEUf4i2goZY-m4wPB0n2zXvxzcEdfTH27iPp-aKiJjfJpYb_ZnHyklk__gZlAy9r7W0594dY-eBJ_iUa5aeDsFS2TIfsfjMJsL8NRWY2noiTw5lsneD8dwvr6N_rYcWoFXDyWXHoRitSSd2iYrB80gbeSOBW0wfKtPxNIZrR0uDhkE8FouS5Pk7QBw412kd43GtrEpAijqn3ne7MNUpCtuNfJ8e_NdGDLTR7CSHhC0jfFlchpIvklF42o216NO-OnyJsjdv1w4_w1ugs61fTHDl8lgBalOjOxauKwIvJJOyFdWmpjlXuzJhrray7ov25uh2ibvFv3Gfd2iuGUnLIZzYBOTT8ftGWTCGXTDvVOvzGbs.c3qMNb2Bne-7g0Wz.PInghFM6Q8Gn0p4Tlebig32s-ZrpLqTMqQDlpXLLYx0iq-StrKco_HrjdN4MxondP4CicCgseIjcV8JR29jKYX-nqKdchEYq_vVIzFHcNI_Mx7y1el192QbMyx6b0Gbj5L79wpuB7qCUqTBNhJZ2c07PuyPsewcNwgIvnc-OrA-2vL6lJnBi5ZGH8gBH1cZCgmbrMpZGNFPG3oFpOn9JPzmnvQxe9tvSFFj5989A8d_XMHP-ZQ.dJZxnBRxJeMKswDsCA3cXA"}	Check yourself by using Private Key included in the response.

User Lifecycle API & SDK

Once Verestro configured a project for your program and you are ready to authenticate with us using JWE token you will need to register users on our platform.

Please check the following components:

- If you want to integrate directly from mobile appliacations or integrate server-to-server - User Lifecycle and Card Management API &SDK.

User registration & KYC

Once you registered users on our platform and would like to create accounts and issue cards for them you need to perform KYC. There are three alternative scenarios:

- You can use the Verestro KYC API in the verification process of your users.
    - Users can register from the level of the mobile application using the SDK method /customers/me/register.
    - You can also use a dedicated method in the server-to-server connection to initiate the verification of your users /secure/customers/(customerid)/register.
- If you already have KYC verification process on your side, just update the KYC flag for the user using User Lifecycle & Card Management API.

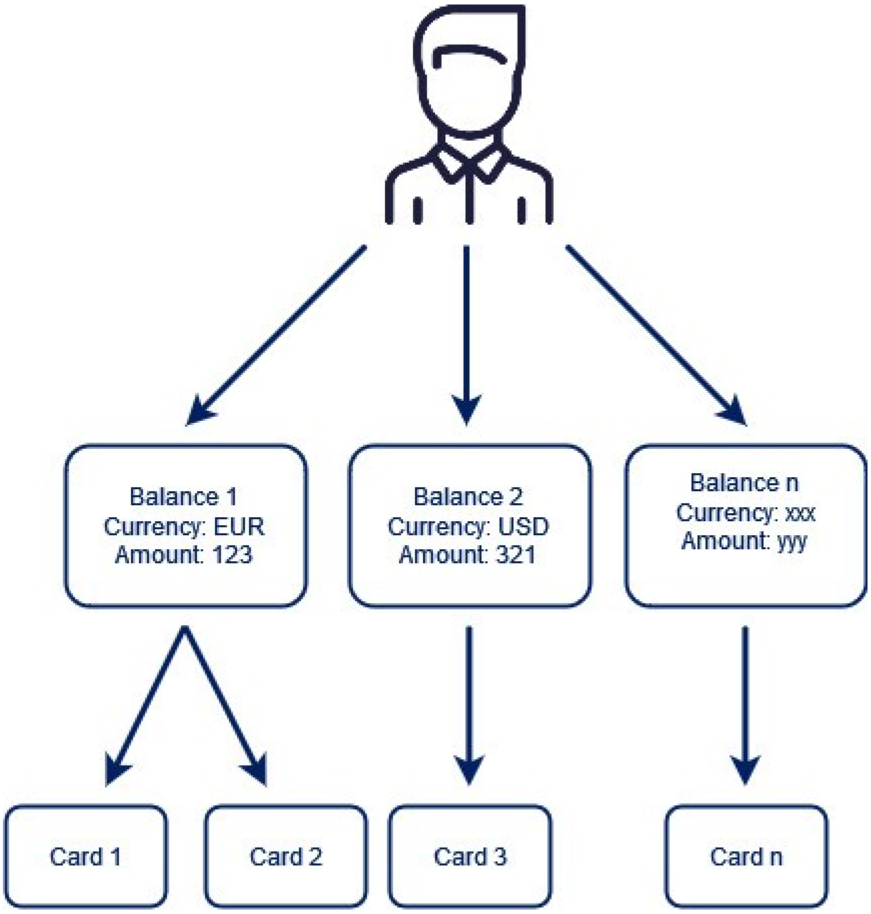
Once you registered users and performed KYC you can initiate account (called "balance") creation.

Create User Balance

It is main account balance that is connected with user account and card. Main User Balance attributes are currency, balance value and balance state. In order to create User Balance make sure user got through KYC process. KYC process can be either manual or automated. It can be performed by partner or Verestro. It is highly recommended that User Balance is hold by Verestro but we can approve projects where partner holds User Balance.

In order to create any payment card at Verestro CMS you have to create User Balance first. Payment card issued for particular User Balance cannot be moved to another balance later.

There is an important rule - one user can have multiply balances and for every balance user can have multiply payment cards.



To create User Balance use the following methods:

- in case of server-to-server connection /secure/customers/(customerid)/balances,
- in case of integration through mobile application /customers/me/balances.

For more information about account / balance management please check technical APIs.

Card issuing

With the Antaca API you can offer your customers three types of cards:

- Virtual card - Digital card without any physical components.
- Physical card – The traditional plastic payment card.

To be able to issue a card for a user, 3 requirements must be met:

- User must exist in a PCI DSS compliant **Data Core** system in Verestro. Make sure you register user via User Lifecycle API & SDK.

- User must be strongly verified according to **KYC.** You can use Verestro KYC or own KYC process.
- The user must have a **User Balance** under which the card will be generated.
- After those 3 steps you can issue a card for the user.

Digital card

If the API receives the request, it will create a 16-digit PAN (Permanent Account Number), CVC2 (Card Verification Code), and Expiry Date. You can then deliver this information to your customer.

```
@startuml
skinparam ParticipantPadding 30
skinparam BoxPadding 30
skinparam noteFontColor #FFFFFF
skinparam noteBackgroundColor #1C1E3F
skinparam noteBorderColor #1C1E3F
skinparam noteBorderThickness 1
skinparam sequence {
ArrowColor #1C1E3F
ArrowFontColor #1C1E3F
ActorBorderColor #1C1E3F
ActorBackgroundColor #FFFFFF
ActorFontStyle bold
ParticipantBorderColor #1C1E3F
ParticipantBackgroundColor #1C1E3F
ParticipantFontColor #FFFFFF
ParticipantFontStyle bold
LifeLineBackgroundColor #1C1E3F
LifeLineBorderColor #1C1E3F
}
actor user as u
participant "mobile app" as m
participant antaca as a
participant datacore as d
participant "payment processor" as t
u->m: 1. generate card
m->a: 2. generate card(userID, SaldoID, configuration ID)
a->t: 3. generate card(cardholder, terminal)
t-->a: 4. card data
a->d: 5. store card
d-->a: 6. status
a-->m: 7. status
@enduml
```

Physical card

```
@startuml
skinparam ParticipantPadding 30
skinparam BoxPadding 30
skinparam noteFontColor #FFFFFF
skinparam noteBackgroundColor #1C1E3F
skinparam noteBorderColor #1C1E3F
skinparam noteBorderThickness 1
skinparam sequence {
ArrowColor #1C1E3F
ArrowFontColor #1C1E3F
ActorBorderColor #1C1E3F
ActorBackgroundColor #FFFFFF
ActorFontStyle bold
ParticipantBorderColor #1C1E3F
ParticipantBackgroundColor #1C1E3F
ParticipantFontColor #FFFFFF
ParticipantFontStyle bold
LifeLineBackgroundColor #1C1E3F
LifeLineBorderColor #1C1E3F
}
actor user as u
participant "mobile app" as m
participant antaca as a
participant "payment processor" as t
participant "card personalization institution" as ac
u->m: 1. order card
m->a: 2. OrderCard(delivery address, userID, SaldoID, configuration ID)
a->t: 3. OrderCard(cardholder, delivery address, terminal)
t-->a: 4. status
a-->m: 5. status
t->t: 6. GeneratePAN and prepare binary file
t->ac: 7. order card
t->t: 8. Generate orderCardReport
a->t: 9. get orderCardReport
t-->a: 10. orderCardReport
a->a: 11. connect card with user and saldo
a->t: 12. linkCard(trackingNo, reference)
t-->a: 13. status
a->t: 14. getAllLinkedCards
t-->a: 15. full card data
a->a: 16. store card in DC
ac-->u: 17. delivery card
```

```
u->m: 18. activate card
m->a: 19. activateCard
a->t: 20. activateCard
t-->a: 21. status
a->t: 22. update PIN (wPIN)
t-->a: 23. status
a->a: 24. update status in DC
@enduml
```

Actions

Create virtual

This method enables creation of virtual payment card for already created user and balance.

Availability

Collection	URL	Authentication	Encryption required	Available for admin roles
Customer	**POST** /customers/me/cards/virtual or for an asynchronous process **POST** /customers/me/cards/virtual/async	Session token	YES* - JWE *for an asynchronous process NO	N/A
Admin	**POST** /admin/customers/{customerId}/cards/virtual	Session token	YES* - JWE	Admin, Manager
API	**POST** /secure/customers/{customerId}/cards/virtual or for an asynchronous process **POST** /secure/customers/{customerId}/cards/virtual/async	x509 certificate	YES* - JWE *for an asynchronous process NO	N/A

Lock

This functionality enables temporary or fixed blocking of already issued cards. After card being blocked every authorisation request will be rejected. While using this method you need to inform CMS Antaca about reasons of card blocking. List of reasons is described below in the table.

Code No	Card stop reason	Failure Action code on POS/ATM	irreversible
1	Card lost	2008	YES
2	Card stolen	2009	YES
3	Pending query	1000	NO
4	Card consolidation	1016	NO
5	Card inactive	1018	YES
6	PIN tries exceeded	1006	NO
7	Suspected fraud	1002	NO
8	Card replaced	1011	YES

Availability

Collection	URL	Authentication	Encryption required	Available for admin roles
Customer	**POST** /customers/me/cards/{cardId}/lock	Session token	NO	N/A
Admin	Administrator blocks the card through the core of the administration panel	N/A	N/A	N/A
API	**POST** /secure/customers/{userId}/cards/{cardId}/lock	x509 certificate	NO	N/A

Unlock

This functionality enables unblocking previously blocked card. It works in case the card was not blocked with Code No 1, 2, 5 or 8 described in the above table (card lost, card stolen, card inactive or card replaced). CMS Antaca does not need reasons for card unblocking.

Availability

Collection	URL	Authentication	Encryption required	Available for admin roles
Customer	**POST** /customers/me/cards/{cardId}/unlock	Session token	NO	N/A
Admin	Administrator unblocks the card through the core of the administration panel	N/A	N/A	N/A
API	**POST** /secure/customers/{userId}/cards/{cardId}/unlock	x509 certificate	NO	N/A

Remove

This functionality enables card deletion from CMS Antaca. Deleted card cannot be restored.

Availability

Collection	URL	Authentication	Encryption required	Available for admin roles
Customer	**DELETE** /customers/me/cards/{cardId}	Session token	NO	N/A
Admin	Administrator delete the card through the core of the administration panel	N/A	N/A	N/A
API	Other APIs remove the card via LC or directly in DC	N/A	N/A	N/A

Get full data

This functionality enables receiving full card data (PAN, Expiry Date, CVC2 or CVV). Access to those data for user should be always connected with additional authorisation by user (fingerprint, application PIN).

Availability

Collection	URL	Authentication	Encryption required	Available for admin roles
Customer	**GET** /customers/me/cards/{id}	Session token	YES	N/A
Admin	Administrator cannot view the full details of the cards	N/A	N/A	N/A
API	**GET** /secure/customers/{customerId}/cards/{id}	x509 certificate	YES	N/A

Reset CVV

This functionality enables generation of new CVC2 or CVV number for virtual.

Availability

Collection	URL	Authentication	Encryption required	Available for admin roles
Customer	**POST** /customers/me/cards/{cardId}/cvv	Session token	YES	N/A
Admin	**POST** /admin/cards/{cardId}/cvv	Session token	YES	Admin, Manager, Employee
API	N/A	N/A	N/A	N/A

Order physical card

This functionality enables ordering plastic card. Process of card personalisation can take up to 48 hours depending on chosen personalisation center. Additionally card will be transferred to user by courier or post office. Physical card ordered by this functionality will be inactive until activation action.

> The DEV/BETA environment does not support physical card order testing.

Availability

Collection	URL	Authentication	Encryption required	Available for admin roles
Customer		Session token	YES	N/A
Admin		Session token	YES	Admin, Manager, Employee
API		x509 certificate	YES	N/A

Link card

Around 48 hours after card ordering it will be visible in user resources. After Verestro receives confirmation from personalisation center that card was personalised CMS Antaca connects card with user account and balance. From this moment it can be visible for user and can be activated.

Set PIN

This functionality is available for physical and virtual cards. It enables setting up PIN that is used for face-to-face transactions (POS and ATM). In the case of virtual cards - for ATM withdrawals.

IMPORTANT:

- After setting up new PIN it is required to perform standard chip & pin transactions (recommended on ATM) to transfer PIN to chip on the plastic to be able to process off-line PIN transactions.
- Majority of POS terminals verifies offline PIN what can result in message "Incorrect PIN" on terminal. User should be informed about it.
- In case of contactless transactions online PIN will be used in all cases so user will not receive "Incorrect PIN" message on terminal.

Availability

Collection	URL	Authentication	Encryption required	Available for admin roles
Customer	**POST** https://prepaidapi.upaid.pl/customers/me/cards/{cardId}/pin	Session token	YES	N/A
Admin	N/A	N/A	N/A	N/A
API	N/A	N/A	N/A	N/A

Activate card

This functionality enables activation of previously ordered physical card. Card transactions will not work until card is activated.

Availability

Collection	URL	Authentication	Encryption required	Available for admin roles
Customer		Session token	NO	N/A
Admin		Session token	NO	Admin, Manager, Employee
API		x509 certificate	NO	N/A

Lock outside

This functionality enables blocking of card in CMS Antaca on request of external entities (MC or VISA or acquirers). It can be used in case user entered incorrect PIN 3 times or in other fraud related actions. This lock cannot be removed if card was blocked by Code No 1, 2, 5, 8 (see below). The table below contains all possible reasons of card lock.

Code No	Card lock reason	Failure Action Code on POS/ATM	Irreversible
1	Card lost	2008	YES
2	Card stolen	2009	YES
3	Pending query	1000	NO
4	Card consolidation	1016	NO
5	Card inactive	1018	YES
6	PIN tries exceeded	1006	NO
7	Suspected fraud	1002	NO
8	Card replaced	1011	YES

More information on Partner Balances and Deposit Requirements

Partner Balance

The partner balance is used in Verestro deployments together with the partner and BIN sponsor. The partner balance secures the financial liquidity of the BIN sponsor in the settlement process, while giving the partner the opportunity to manage the balances of its users.

Partner Credit Balance

Partner Credit Balance is used to process transactions of Partner especially in cases where User Balance is hold by Verestro. Examples of such projects are many standard projects where Partner is not financial institution or e-Wallet and does not hold User Balances on its side.

The main reason to use Partner Credit Balance is limiting transactions performed by Partner's users to funds hold on Partner Credit Balance. Verestro and its BIN sponsors cannot risk processing transactions without having funds available so this deposit needs to be used to enable transactions in such cases.

Partner through Verestro Administration Panel has access to actual level of Partner Credit Balance and can reload it by sending banking transfer to BIN Sponsor cooperating with Verestro. Partner can receive notification via e-email if Partner Credit Balance goes below pre-defined level.

```
@startuml
skinparam ParticipantPadding 30
skinparam BoxPadding 30
skinparam noteFontColor #FFFFFF
skinparam noteBackgroundColor #1C1E3F
skinparam noteBorderColor #1C1E3F
skinparam noteBorderThickness 1
skinparam sequence {
ArrowColor #1C1E3F
ArrowFontColor #1C1E3F
ActorBorderColor #1C1E3F
ActorBackgroundColor #FFFFFF
ActorFontStyle bold
ParticipantBorderColor #1C1E3F
ParticipantBackgroundColor #1C1E3F
ParticipantFontColor #FFFFFF
ParticipantFontStyle bold
LifeLineBackgroundColor #1C1E3F
LifeLineBorderColor #1C1E3F
}
participant Partner as p
participant "Issuer Bank" as b
participant "Licensed Issuer" as i
participant Antaca as a
actor "End users" as u
p->b: 1. bank transfer
b->b: 2. accounting of funds
b-->i: 3. accounting of funds
i->a: 4. top up credit balance
p->a: 5. top up user balance
a->a: 6. charge credit balance
alt
a->a: 7. top up user balance
a-->p: 8. success
else insufficient funds
a-->p: 9. fail (insufficient funds)
end
@enduml
```

Partner Deposit Balance

Partner Deposit Balance is used alternatively to Partner Credit Balance. Partner Credit Balance is used to process transactions of Partner especially in cases where User Balance is not hold by Verestro. Examples of such projects are the ones with other wallet providers that already hold user balance or project where Verestro through its partners acts as BIN Sponsor or Principal Member for Affiliate Partner.

The main reason to use Partner Deposit Balance is limiting transactions performed by Partner's users to funds hold on Partner Deposit Balance. Verestro and its BIN sponsors cannot risk processing transactions without having funds available so this deposit needs to be used to enable transactions in such cases.

Partner through Verestro Administration Panel has access to actual level of Partner Deposit Balance and can reload it by sending banking transfer to BIN Sponsor cooperating with Verestro. Partner can receive notification via e-email if Partner Deposit Balance goes below pre-defined level.

```
@startuml
skinparam ParticipantPadding 30
skinparam BoxPadding 30
skinparam noteFontColor #FFFFFF
skinparam noteBackgroundColor #1C1E3F
skinparam noteBorderColor #1C1E3F
skinparam noteBorderThickness 1
skinparam sequence {
ArrowColor #1C1E3F
ArrowFontColor #1C1E3F
ActorBorderColor #1C1E3F
ActorBackgroundColor #FFFFFF
ActorFontStyle bold
ParticipantBorderColor #1C1E3F
ParticipantBackgroundColor #1C1E3F
ParticipantFontColor #FFFFFF
ParticipantFontStyle bold
LifeLineBackgroundColor #1C1E3F
LifeLineBorderColor #1C1E3F
}
participant Partner as p
participant "Issuer Bank" as b
participant "Licensed Issuer" as i
participant Antaca as a
participant "Payment cloud" as mc
participant "POS/ATM" as pos
actor "End users" as u
p->b: 1. bank transfer
b->b: 2. accounting of funds
b-->i: 3. accounting of funds
i->a: 4. top up deposit balance
u->pos: 5. make payment
pos->mc: 6. payment authorization
mc->a: 7. payment authorization
a->a: 8. charge deposit balance
alt Sufficient deposit funds
a->a: 9. lock user funds
alt Sufficient user funds
a-->mc: 10. authorization success
mc-->pos: 11. authorization success
pos-->u: 12. success
else Insufficient user funds
a-->mc: 13. authorization fail
mc-->pos: 14. authorization fail
pos-->u: 15. fail
end
else Insufficient deposit funds
a-->mc: 16. authorization fail
mc-->pos: 17. authorization fail
pos-->u: 18. fail
end
@enduml
```

Balance Summary in Administration Panel

Summary Balances are a control tool used for accounting and liquidity verification reasons. They are presented in Administration Panel in every currency used in the project.

Users

Presents sum of all User Balances in particular currency.

Wallet

Presents sum of all User Balances and all Partner Balances in particular currency.

Actions

Create user balance

This functionality enables creation of user balance in particular currency.

Create Partner Deposit Balance or Partner Credit Balance

Not used in standard projects. This functional enables Partner creation of new Partner Deposit Balance or Partner Credit Balance for particular projects.

Get User Balance

Enables getting user balance and list of cards connected to this balance (account).

Get Partner Deposit Balance or Partner Credit Balance

This functional enables Partner getting information of Partner Deposit Balance or Partner Credit Balance for particular projects.

Reload Partner Deposit Balance or Partner Credit Balance

Not used in standard project. This functional enables Verestro to reload Partner Deposit Balance or Partner Credit Balance for particular projects. It is used by Verestro.

Reload user balance

This functionality enables reloading User Balance.

Fee management

Fee Management System documentation: Fee Management Platform | Verestro Developer Zone.

It is possible to setup various fees charged to users for card issuing and account management activities. Fees can be setup through administration panel by customer or dedicated Verestro customer services. Fees can be managed in two ways:

1. Partner can setup own fee management system and charge users completely outside of Verestro system
2. Partner can use Verestro fee management module available in Administration Panel

There are various fees that can be configured via Administration Panel:

- fee for account creating
- fee for card creating
- ~~fee for token creating~~
- monthly / weekly / daily fee per card
- ~~monthly / weekly / daily fee per account~~
- POS transaction fees (fixed and percantage)
- eCom transaction fees
- ATM transaction fees
- Money transfer fee (IBAN Transfer)
- Currency conversion fees
- and others

There is implementation on-going to have conditional fees like - "*if users do 1000 eur transaction monthly we do not charge monthly fee*".

Please consult Verestro sales or Project Manager in case you need more information.

Other functionalities

You can find additional methods in API descriptions:

- API used for server-to-server connections,
- API used for mobile application-to-server connections,
- API used for Administration Panel access (rarely used by partners).

In case of questions please let us know.

Revision #91
Created 2 April 2022 14:02:39 by Tadeusz Krysa

Updated 30 December 2024 09:22:34 by Barbara Tudruj