

# Overview

Puran is intended to be a product that provides comprehensive functionality for the mobile end user. The available range of functionality is wide and can be freely modified according to product requirements (for example, a given customer does not want NFC but wants Apple Wallet and Google Pay support). This is made possible by its modular design. Puran's capabilities can be expanded at will using and third-party client SDKs. There is nothing preventing the client API functionality from being implemented as a new module.

Puran's basic capabilities are to provide a consistent and rapid end-user interface to:

- registration and authentication,
- user and device pairing,
- mobile application security support with full biometrics support (including in the form of Apple TouchID and Apple FaceID),
- support for digitizing and tokenization of payment cards,
- support for a wide range of payments and money transfers (e.g. NFC, Peer2Peer, QR, PayByLink),
- eKYC mechanism,
- support for transaction history with possibility to add attachments such as invoices, receipts, payment confirmations,
- redeeming of corporate cards,
- storing loyalty cards,
- creation of user balance and virtual cards.

## Purpose and scope

This product guide provides a high-level overview of Puran product. This document covers the following topics:

- description of possible configurations,
- granting access,
- description of main processes as,
- additional and optional functionalities.

## Terminology

This section explains a number of key terms used in this document.

Name	Description
End-user/User	User using the mobile application.
SDK	Software development kit in the form of application programming interfaces (APIs) libraries of reusable functions used to interface to a particular programming language (Swift and Kotlin).
MDC SDK	Mobile Data Core SDK.
VCP SDK	Verestro Cloud Payments SDK.
P2P SDK	PeerToPeer SDK.
BC SDK	Business Control SDK.

# Available modules

This section of the documentation briefly describes the various available modules that can be enabled within the Puran product. The last part presents the solutions that are currently being produced or are planned to be produced and will be available soon. A more detailed presentation of the various functionalities is presented in the section describing the screens in the application.

## MDC - Basic application flow

### Secure connections to the APIs

The most important module responsible for the security of stored and transmitted user data, including personal information, payment cards and addresses. Creates a secure connection to the API based on a time limited session token. Required for other Verestro's modules to work because of providing identity confirmation.

### Basic user flow

Another functionality provided by the MDC module is handling the end user flow, starting with registration, through the process of pairing the device (password login) and ending with user authentication (authorization of the logged-in application with a pin or biometrics).

## Cards management

Using the MDC module, the application can securely download end-user cards from the PCI DSS environment. It is also available to authenticate the card when adding via 3DS (also in V2 version) and to remove all card data from Verestro systems.

## Transaction history

Also using this module it is possible to download the whole transaction history of the logged user, regardless of whether the card still exists in the system (it is possible to download the transaction history for a card that has been removed). Transaction history contains the most important information about each transaction, such as date, amount, status, merchant and optionally attachments.

## Attachments

The next mechanism provided by the MDC SDK is transaction attachment support. It allows up to 5 (5 per transaction) images related to a transaction to be added on the transaction details screen. This allows, for example, to attach a picture of a receipt or a scan of an invoice. The number of 5 attachments was created in order to be able to attach e.g. several photos of a long receipt or a multi-sided invoice. Attachments can be attached directly using the camera or added from the device memory.

## Push notifications

What's more, using this SDK allows enduser to handle push notifications about transactions. When a new transaction is registered in the Verestro system and in the enduser's account, a push notification is sent, which, when clicked, takes enduser to a refreshed list of transaction history

# VCP – digitization, tokenization, NFC payments

## Digitization and tokenization

The two most important functionalities of the VCP module are digitizing and subsequent tokenization. These processes allow the physical card and its information to be digitized and then a token used for payment to be created.

## NFC

NFC payments are contactless payments that use near-field communication (NFC) technology to exchange data between readers and payment devices such as phone using Puran Application. The NFC module allows you to make payments directly using the Puran app. For technological reasons, this functionality is only available on Android phones. Apple iOS can only use Apple Wallet.

# P2P – peer to peer payments, QR payments, PayByLink payments and receivers

## Peer to peer payments

The use of the P2P module allows funds to be transferred between users using the recipient's data. All you have to do is select the card and enter the data of the recipient who is to receive the funds. These functionalities are developed in the Receivers module. For cards issued by Antaca, it is not required to enter CVC when making a transfer. In this case, after authorization, the CVC is retrieved from PCI-DSS beyond the knowledge of the user. For external cards, we do not store the CVC, so the user must enter it in the process of making a money send

## QR payments

The QR module provides the functionality of using QR codes to process payments. Its operation is intuitive and fast, the recipient of the funds on the QR screen enters the amount they wish to receive and the sender, using our application (this is important, you cannot use an external QR code scanner) simply scans the code and confirms the payment.

## PayByLink payments

PayByLink is a module designed to create payment links that can be used for a limited time. On the corresponding screen, you just need to enter the amount and generate the link. Such a payment link can be sent in any way to the sender, who after clicking it will be taken to the screen with the payment confirmation.

## Receivers

The receivers module extends P2P-related functionality. Its only responsibility is to provide a list of contacts from the device memory and mark which users are using our application. This allows us to select the recipient of the payment without entering their data manually.

# Business Control – business cards and alerts

## Corporate cards

This module allows to receive and activate corporate/business cards and use them for payments like any payment card. Such cards are assigned for a specific period of time and have an amount limit. It is also possible to send a request for increasing the amount limit on the card or extending the validity of the card to the corporation from which you received the card.

## Alerts about business cards

Another of the BC module's functionalities is support for business card notifications. There is displayed a list of notifications for corporate card users about such events as updates of regulations, which acceptance is required for further use of corporate cards or feedback about positive or negative decision on application for card limit increase.

# Antaca – cards issuing and eKYC

## Cards issuing

One of the most important functionalities provided by the Antaca module is the ability to create and issue payment cards. End users can issue both virtual and physical cards in the mobile application.

## eKYC

The next functionality that is included in the Antaca module is eKYC. The eKYC solution offers complete remote identity verification and management. Puran implements the end-user part of the eKYC process for providing identity verification documents. It is also possible to use external (customer's) service for eKYC mechanism. This can be done using webview or native views connected to the external API.

## Card limits

The use of the Antaca module allows end users to manage their personal payment card limits. This is a common functionality encountered with online banking. The enduser can modify the limits for NFC, e-commerce or foreign currency payments for a specific payment card in a few clicks. The use of this functionality unquestionably increases the security of the user's funds.

## 3D Secure

If Antaca is used as a card-generating service, it is also responsible for handling the 3D Secure process. Possible options to use are sending the transaction confirmation code via SMS (possible integration with any messaging server provided by the client or use of Verestro services) or push notifications.

# Apple Wallet & Google Wallet

The functionality provided by the Wallets module is consistent regardless of the target platform. Their most important functionalities are the ability to add a card to Apple Wallet or Google Pay and to check if it has been added. The card thus re-processed can be used for payments and functionalities provided by these wallets. It is important to remember that adding it to external wallet does not mean that you cannot use it further in Puran app or pay with NFC (Android only).

## IBAN - receiving and sending

One of the modules available in the Whitelabel mobile solution is IBAN. It allows endusers to use the IBAN numbers of their currency balances. Enduser can check the IBAN number by going to the details of a particular balance. Using them, enduser can receive as well as send funds.

## Puran – user experience, loyalty cards, fitness

### Additional user experience features

Puran provides mechanisms to make it easier for users to use the mobile application. These involve a number of modules, by extending their default functionality to provide a better user experience. For example, it is possible to add a tutorial to the app displayed at startup between the splash screen and the login screen. Another configuration option is to display the application's splash screen instead of the login screen (then clicking on any component of the splash screen takes enduser to the authorization screen). In addition, it is possible to provide the option of copying the CVC and Pan number on the card details (after additional confirmation of identity) so that the user does not have to rewrite them for payments such as e-commerce.

### Loyalty cards

The functionality of handling loyalty cards is provided by the Loyalty module. Currently its operation is strongly connected with device memory (card data is lost when the device is unpaired) but we are working on creating a solution which will store loyalty cards on a server so that user will always have access to them. This module allows you to scan a loyalty card and create its digital form, so that you can always have all your loyalty cards with you, in our application.

### Addresses - calling cards

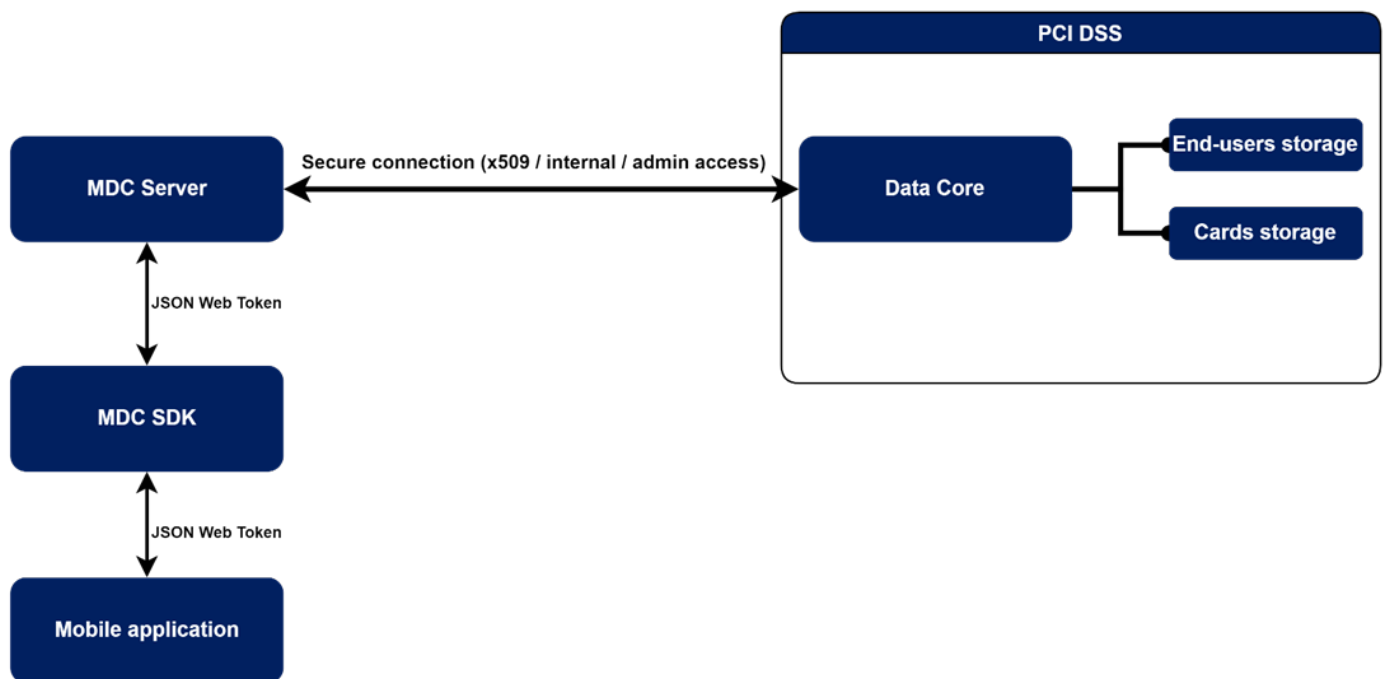
A functionality similar to loyalty cards is the address module. It allows enduser to store in the form of calling cards not only private address or addresses of company branches, but also to add and store data about contractors or important places.

## Fitness

The fitness module allows to pair dedicated sports bands for use during workouts. The data collected in this way can be analysed and help you achieve your sporting goals optimally.

# Security

The systems offered by Verestro are fully secure, which is confirmed by current third-party certificates. As we store card and payment data we are obliged to comply with strict legal requirements. Card data are stored in a specially designed environment - Data Core. This environment is PCI DSS certified. The PCI-DSS standard guarantees the security of payment card data. It ensures that sensitive information is properly guarded and provides maximum security in the payment process.



We achieve high security standards by, among other things :

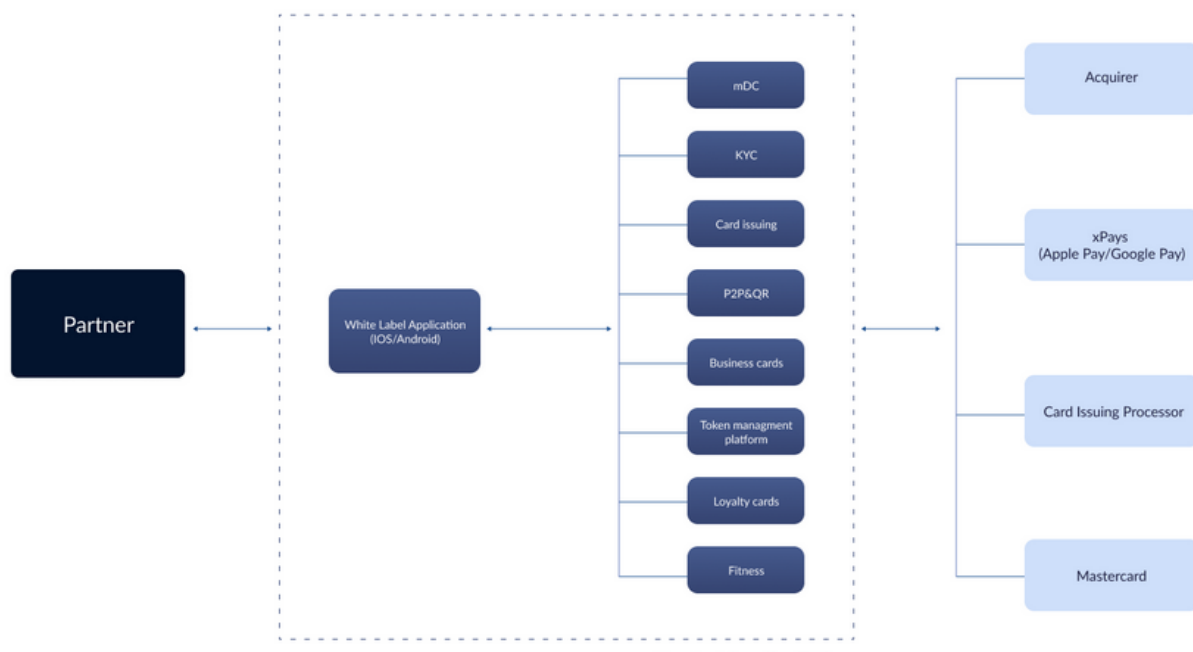
1. Building and maintaining network security - the need to build and maintain a firewall configuration that protects cardholder data, not using manufacturers' default passwords and settings.
2. Protecting cardholder data - protecting stored cardholder data, encrypting data transmissions when using public networks.
3. Maintaining a payment management program - using regularly updated anti-virus systems, developing secure systems and applications.
4. Implementing strong access control methods - limiting access to cardholder data to only those with a business need, assigning each user a unique ID, limiting physical access to cardholder data.

5. Regular network monitoring and testing - testing security systems and processes, controlling access to network resources and cardholder data.
6. Maintaining information security policies - relying on security policies for employees and vendors.

The following example shows the connection between the mobile application and the PCI DSS environment to retrieve the end user's card list.

# Architecture

Puran uses Verestro's distributed systems to provide the highest quality of service. It is practically the best architectural solution these days. As mentioned in the previous chapter, the communication between services is completely encrypted, maintaining the highest security standards. This kind of system guarantees not only high efficiency, due to the division of responsibilities between the components, but also allows for easy and fast scaling of the system according to the customer's requirements. The state of connectivity between the various parts of the system is constantly monitored, ensuring immediate response to any out of the norm events. We also use tools for storing and processing logs, thanks to which we can provide high quality support and solve any issues in a short time.



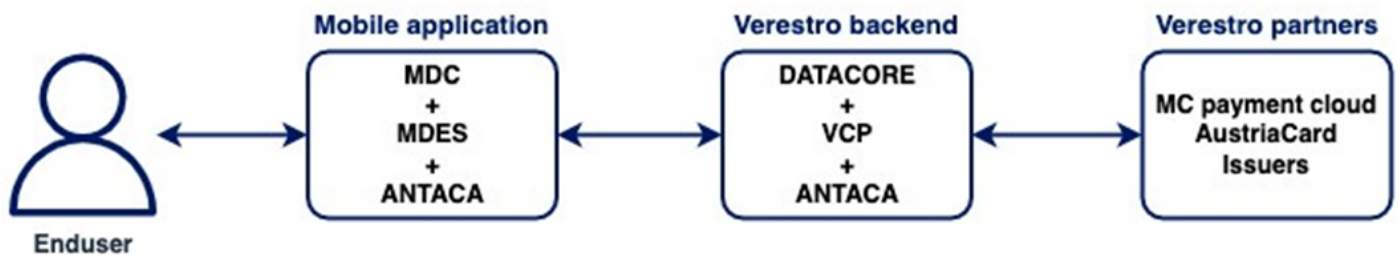
## Access solutions



This chapter describes possible implementations of the Puran product. We are not limited to developing an entire application, we can provide a customised product based on client requirements and vision.

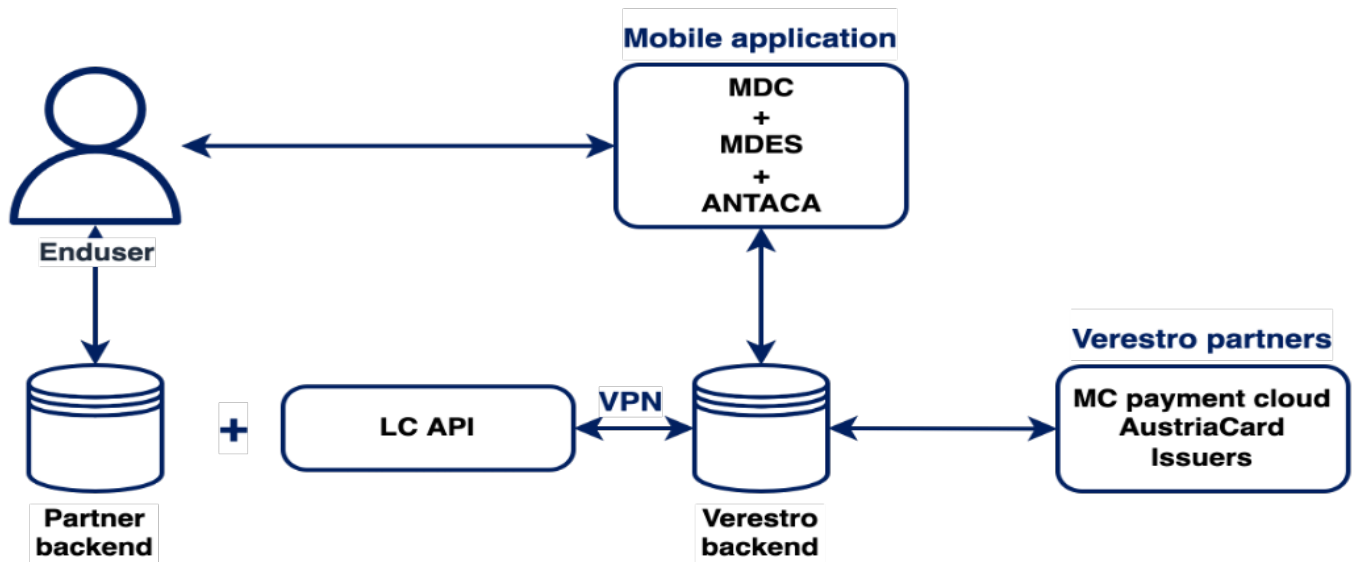
## Comprehensive solution

The first possible solution for delivering the product is a closed and finished application. This process starts with the creation of a prototype application for the client depending on the modules selected and the branding provided. This solution is entirely contained in the services provided by Verestro. No technical work is required on the customer side.



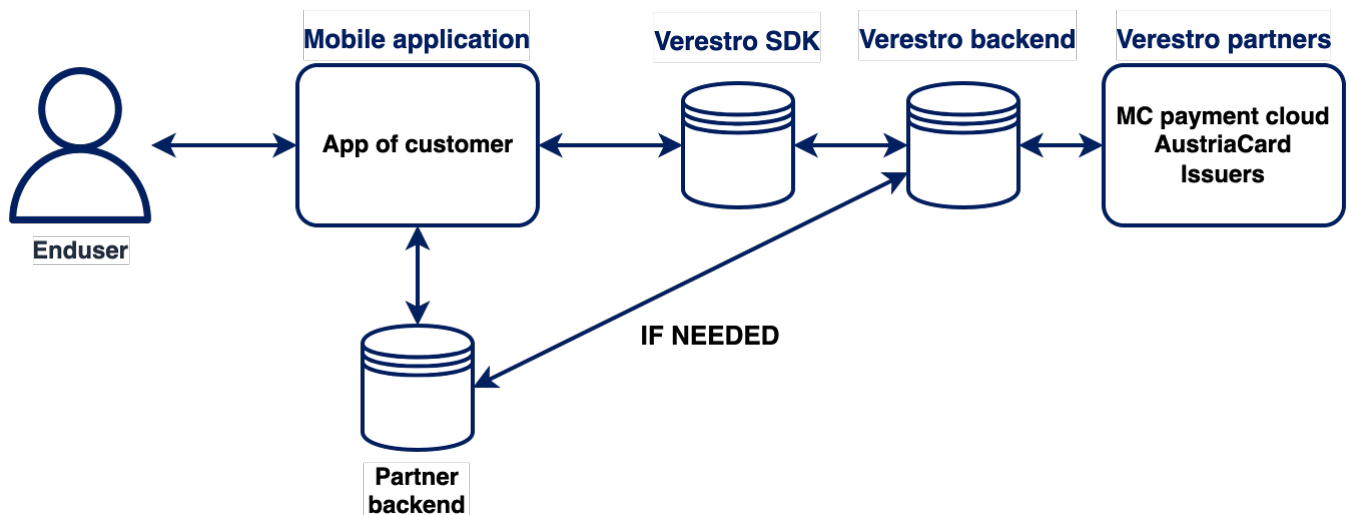
## Application connected to the existing infrastructure

If the customer has already built a working product with its own users, but would like to extend its offer with mobile applications with selected modules, there is also such a possibility. Thanks to the use of a dedicated LC API, it is possible to inject users' mobile applications into the backend to avoid forcing them to re-register. The application is designed and implemented on the basis of the white label in accordance with the client's expectations.



## SDKs

Another implementation option is the use of a package of selected SDKs for application. With such an implementation, the use of MDC is always required to ensure the security of data transferred to the backend of selected SDKs. By choosing this option, there is no need to integrate user databases or to set up new accounts by endusers.



## Configuration

The following section contains Puran product configuration options that are available by default and can be implemented easily in the application development flow.

# Appearance of the application

By default, the application allows to change the branding, which includes:

- splash screen (occurs after application launch, before any other screen),
- colors of application (primary, secondary and accent),
- text appearance (size, font, color),
- icons and logos,
- name (on device and on market),
- visuals of cards.

Any other change, whether in terms of design or flow/functionality, requires analysis by the technology team.

For example, adding additional screens during adding a card or additional data fields that are available on the registration screen and then presented in the application.

# User registration and activation process

Possible configuration options for processes related to application access.

Registration	Description
None	The application is open for use in full or limited form without login. This option is not available for some functionalities (especially related to payments and cards).
Open for everyone	Use of the application is possible after registration, which is open to all.
Invitation required	Use of the application is possible after registration, which is available only by invitation code.

Activation	Description

None	Once registration is complete, no confirmation is required to activate account.
E-mail	Once registration is complete, a link is sent to your email address. Its click takes you to a web page (fully configurable HTML) and calls a callback to the server, which will activate your account. Web page can simply contain information about correct account activation or contain content introducing user to the application.
SMS OTP	Once registration is complete, an SMS is sent with the OTP code that is required to activate your account. If the user disables the OTP window during the registration process (e.g. by turning off the application), it will be called again during the login attempt.

eKYC	Description
Enabled	Enabling the eKYC process may be required by some functionality or for legal reasons. Enabling this option forces the user to go through the KYC process through Verestro's internal process - taking a photo of their face and identity document with the mobile device's camera.
Disabled	The application does not require a KYC process to use any of the features .

# Time settings for individual functionalities

Puran has a several default parameters related to the time of each action. Table below describes particular action and time related to the action.

Functionality	Description	Default time on beta environment	Default time on production environment
Session time	Session after successful login to the mobile application.	15 minutes	15 minutes

Cache lifespan	Specifies how long the cache is considered up-to-date.	15 minutes	15 minutes
----------------	--	------------	------------

# Storage of specific data

This section describes the storage locations for information that is used in the mobile application.

Functionality	Description
User credentials	Only as JWT token. They are not stored in a direct way.
Identity token	In secured form, one token per user. Logging out removes the token.
Cards data	As a cache with a specified lifespan in the local database per logged in user. All card data is stored in a PCI-DSS environment.
Loyalty cards	Currently in local device memory, so logging out deletes loyalty cards. In the future they will be associated with the user like payment cards and downloaded from the server.

# Requirements for sensitive data

This section contains password and pin detailed requirements. Password has to contain at least 3 of the 4 groups of characters mentioned below.

Functionality	Description
User password length	8-255 chars.
Password requirements	upper-case letter, lower-case letter, special character and digit.
User PIN length	4
PIN requirements	only digits.

# Permissions

This section contains a list of required permissions and the reasons for using them.

Functionality	Description
---------------	-------------

Camera	Camera permissions are required for the following processes: eKYC, QR and transaction attachments.
Storage & multimedia	Memory access is required in the process of transaction attachments.
Contacts	Access to contacts is required in the P2P process to download a list of contacts that can be used as recipients of a transfer.

# Cache management

In order to provide the user with an application that runs quickly and smoothly, a cache mechanism has been implemented for the most frequently used data. Some card data, transaction history or loyalty cards are cached in a secure way.

Cache memory is designed to store data that will be processed by the system in a short time. Its main advantage is the speed of writing and reading, so the role it plays determines the performance of the device. This happens, among other things, thanks to the small capacity of the medium. Well, the smaller the amount of space, the shorter the waiting time to find a particular unit. Additionally, cached data is available immediately, regardless of the server response and is refreshed every specified time by special mechanisms in Puran.

Cache settings are not global, they can be modified per resource, e.g. separately for cards and transaction history.

When modifying data that is contained in the cache, the cache is refreshed, e.g. when card data is modified.

For technological reasons, this mechanism differs between the Google Android and Apple iOS platforms. The following chapters provide an overview of the possible caching options for each execution platform.

## Google Android

Cache strategy	Description
CachedFirst	Tries to get cached data respecting cache status, if cache fetch fails (or cache validity expired) then tries to get data online.
OnlineFirst	Tries to get data online first, on fail can throw exception or force fetch data from cache (not respecting cache status), configurable by cachedOnFail argument.
ForceCached	Gets data from cache not respecting cache status.

There are some exceptions in caching e.g. transaction history is currently always loaded from cache and refreshed only when cache expires.

## Apple iOS

The iOS MobileDCSDK provides a functionality of persistent caching of data which allows the SDK to store recently fetched user data, such as cards, addresses etc. It is beneficial to store the data in case of possible future failures while fetching due to a lack of internet connection or server errors so a user may still be able to access his data.

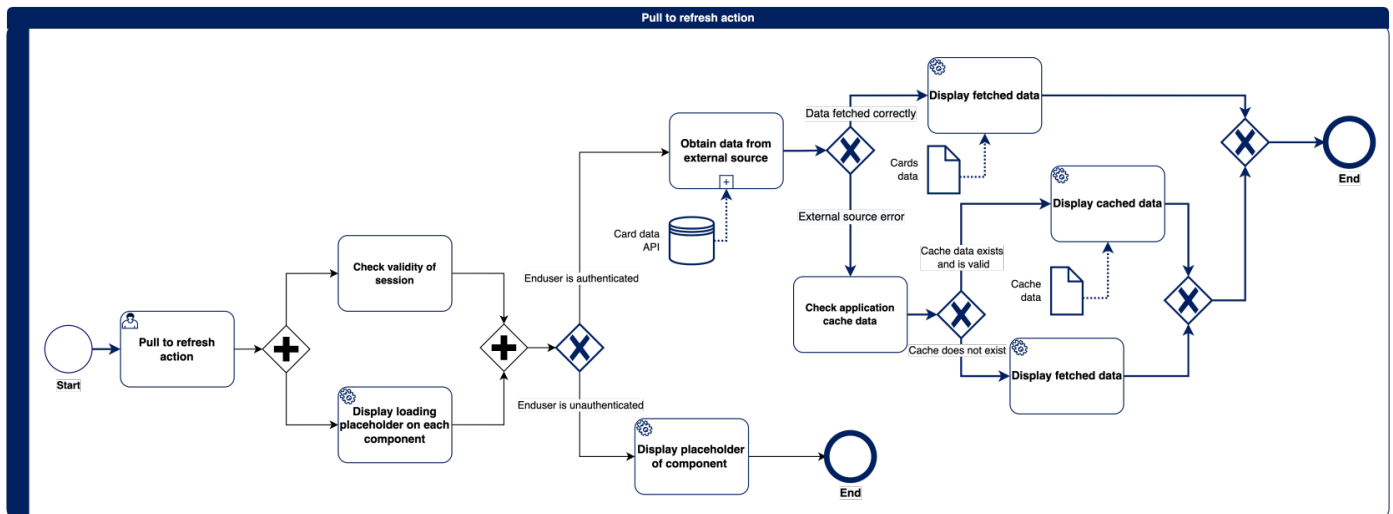
*Fetching Strategy* determines how methods, marked with **Cacheable**, should behave. That means they can be used to fetch the data only online, offline (from cache) or by mixing online fetching with cached data as an alternative in case of a failure. Strategy can take one of the following values:

Cache strategy	Description
Online	Sets fetching to be performed online, without caching. It disables caching globally, which means that even when using other strategies for a specific method, the data will <b>not</b> be cached. Further description can be found below.
OnlineFirst	Sets fetching to be performed online, unless a response failure occurs. If so, data will be retrieved from the cache. Each online response is saved to the cache.
FromCache	Sets fetching to be performed offline by getting the data directly from the cache. When there is no cache, the data will <b>not</b> be fetched from the API.
CacheFirst	Sets fetching to be performed offline, retrieving the data from the cache, if the data in cache is valid. It means that the data will be fetched from the cache unless it is invalidated.

## Manual cache refresh mechanism - pull to refresh

Our mobile system allows the user to manually force the cache to be refreshed. To carry out such an operation on the home screen, pull down the upper edge of the screen and perform the "pull to refresh" gesture well known to users of mobile devices. At this point, the current information is downloaded from all external data sources (cards, balances, notifications, etc.). If the current data has been successfully downloaded, the cache is replaced by the downloaded information. Otherwise, the data that were in the cache before the refresh attempt are displayed. If the cache

does not exist or its validity has expired, the components for which no information is available are displayed. Clicking on such a graphic results in displaying data for that particular component (e.g. card list or balance). It is possible to perform the "pull to refresh" operation again to refresh all the data.



# Application delivery for testing and production purposes

Completion of product configuration (T&C regulations, branding, test groups) is required to test mobile applications.

For beta environment testing, it is necessary to provide the project manager information about the email address of tester and device type . This is related to separate app delivery solutions for each platform.

In the case of a production environment, the application is provided by authorized and official application stores dedicated to that environment.

## Beta environment

In the initial stages of the project, the mobile application can be delivered as an .APK file to be installed manually on the device. It is also possible to set up an automatic distribution center for test versions, in which case it is enough to provide Verestro with a list of email addresses to which invitations to the test system will be sent. Each user will receive an individual registration link and AppTester software (a fully secure component of the Google Firebase system) or TestFlight software (Apple's standard way to distribute test applications that meet the latest functional and security requirements). Both of the distribution ways allow to download each version of the



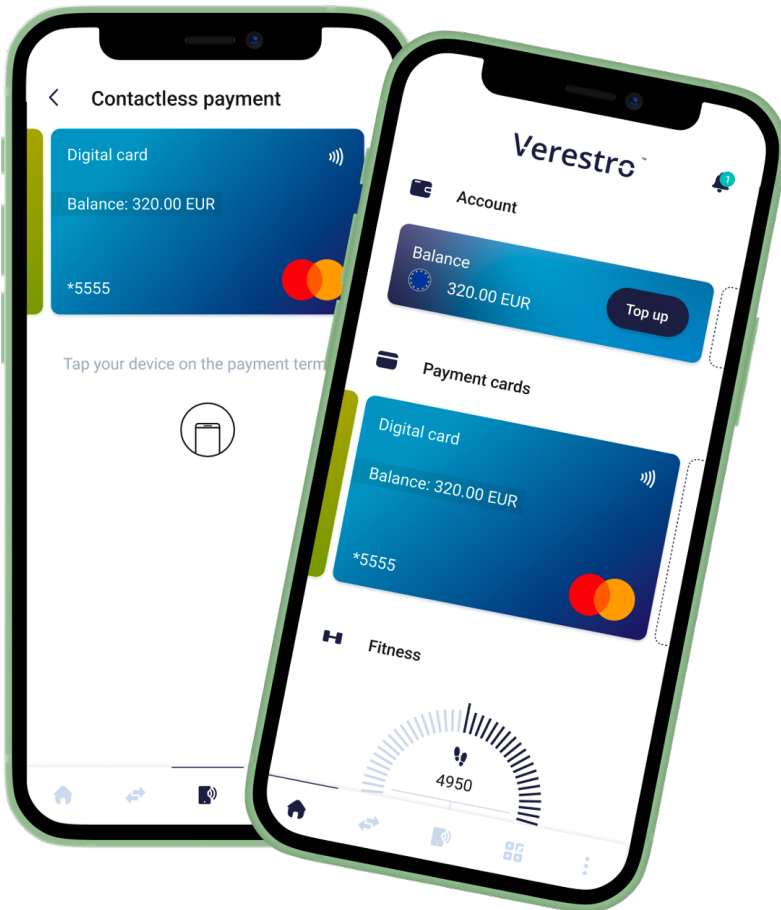
application and deliver new versions in real time to testers.

## Production environment

Once the testing phase is complete, Verestro generates applications that must be signed with the appropriate set of keys and then, using procedures appropriate to the specific distribution site (Apple AppStore or Google Play), added to the app stores. Once the application is in the store, any user can easily and quickly install the application and update it automatically.

## Prototype

This section contains a link to the always up-to-date prototype. Using it allows you to get to know the presented product better and presents its advantages.



---

Revision #61

Created 19 May 2022 09:02:09 by Wiktor Kowalczyk

Updated 9 January 2025 09:51:28 by Wiktor Kowalczyk