

Article

You can find more knowledge about products on this site.

- [Tokenization and Contactless Payments - Verestro's competitive advantages](#)
- [Tokenize it!](#)
- [Push Provisioning Step by Step](#)
- [Push Provisioning & Web Push Provisioning](#)
- [Tokenization in eCommerce Payments](#)
- [Web Push Provisioning - Overview](#)

Tokenization and Contactless Payments - Verestro's competitive advantages

The most important reason why you should use the Verestro Tokenization and Contactless Payment Solutions

There are many reasons why to use [Tokenization](#) and **Contactless Payment solutions** provided by Verestro but one of them is definitely the most important. We can **give you access to ALL ways of NFC** and contactless payments through a single platform and one vendor. There are multiple partners you will have to integrate with and certify. Some processors will enable the majority of them but very few will give you access to all. Some IT companies will tell you that they can do everything and connect to everyone but it is also not true. It takes time and money to get new integrations and certifications.

Examples of questions that you can ask your vendors

- Imagine that 10% of your users are using Huawei phones. Are you ready to provide contactless payments to their phones now?
- What if many of your users have cheap Android phones? Will your solution work on this? How many MB are used by your system (in our case below 4MB)?
- What if you are issuing cards in multiple markets? Is your current solution ready globally?
- What if you are planning to tokenize bank accounts (not cards but bank accounts) - is your platform ready for this?

Verestro's solutions

At [Verestro](#) we solve this problem by **certifying and standardising integration** with all possible ways of payments. We are delivering our tokenization and NFC cloud payment platform globally to multiple bank and payment institutions. Current list of certified solutions is:

- phones: Apple, Huawei, Pixel, Samsung, LG, Motorola, Xiaomi etc.
- operating systems: iOS, Android
- X-pays: Apple Pay, Google Pay, Samsung Pay, Fitbit Pay, Garmin Pay, your own wallet, your own X-Pay
- payment schemes: VISA, Mastercard, Pay-by-Account, Local scheme
- countries: global, possibility of local data hosting or transaction processing

Business benefits

Additional business benefits that make us unique are:

- possibility of registering individual card visuals
- delivering the full scope of reporting for Apple and Google inside the Verestro Token Management Platform
- possibility to organize card registration to Apple or Google with own, proprietary CVC / CVV verification method (no need for integration)
- cheap and quick - below 20k eur one-time, below 4k eur per month, 2-4 months only

Please contact us for more information.

Tokenize it!

Tokenization in Payments – How It Is Transforming Transactions

Tokenization has played a crucial role in contactless payments for years, evolving into a broad concept covering various technologies. Initially, it relied on hardware-based solutions like chips or SIM cards. Today, most systems use a device's standard processor, eliminating the need for dedicated hardware.

Every innovation in this field must comply with payment industry standards. Beyond implementation, security and formal approvals are equally critical. While Mastercard and Visa shape the market, tech giants like Apple and Google also drive payment evolution.

Regional Challenges and Service Development

Not all regions have equal access to payment services. Countries impose regulations requiring localized data processing. India mandates domestic storage of payment data, while Turkey requires accreditation from its central bank (CBRT) for external payment providers.

Apple Pay is widely available in regions like the **United States and Europe**, as well as in other major markets such as **South Korea, Japan, and Australia**. However, its availability is still limited in many parts of the world, meaning that a significant number of iPhone users do not yet have access to contactless payments. By opening its ecosystem to external wallets, Apple creates new opportunities for banks and payment providers. Alternative mobile payment solutions on iPhones are now a reality.

Tokenization in E-Commerce and Mobile Applications

Tokenization extends beyond contactless payments, increasingly used in e-commerce and in-app transactions. Systems like DSRP (Digital Secure Remote Payment) and Click-to-Pay enhance convenience and simplify payments for merchants, particularly those avoiding complex PCI DSS procedures or adapting to regional regulations.

Wearable Payments – What's Next?

Wearable payments are rapidly evolving. Smartwatch payments are common, while rings, glasses, and even clothing are gaining interest. The fintech industry will determine how quickly these

innovations become mainstream. Soon, paying for coffee with a smart ring or VR glasses could be as easy as using a phone.

Tokenization not only simplifies transactions but also shapes the future of finance. The coming years promise even more advancements in this space.

Push Provisioning Step by Step

Android - Push Provisioning to Google Pay

1. Request Access

Before enabling Push Provisioning in the Android application, you'll need to be whitelisted by Google. To initiate this process:

- Visit the **Google Pay Push Provisioning** launch page:
<https://developers.google.com/pay/issuers/apis/push-provisioning/android/launch-process>
- Click "**Request access**" and complete the provided form.

2. Provide Application Details (Upon Approval)

Once your access request is approved, Google will contact you to gather additional information about your application:

- **Screenshots:** Submit screenshots that showcase the placement and functionality of the "Google Pay" button within your app.
- **User Flow Recording:** Provide a screen recording demonstrating the user experience for adding a card to Google Pay.

Important Note: Google Pay integration does not require any additional paid certifications.

3. SDK Configuration

- **Verestro Application Ownership:** If Verestro develops the application, our development team will handle the internal configuration of 3rd party Google Pay SDK.
- **Customer Application Ownership:** If the customer owns the application, customer's development team needs to get the Google Pay SDK and follow official documentation.

iOS - Push Provisioning to Apple Pay

1. Contact Apple

To begin enabling Push Provisioning for **Apple Pay**, initiate contact with Apple directly. Use the following email addresses:

- apple-pay-provisioning@apple.com
- applepayentitlements@apple.com

2. Application Information and Testing

Provide Apple with the details of your App Store application. Verestro will conduct integration testing using the same application (including the same ADAM ID) within the TestFlight platform.

3. FIME Certification (Optional)

While not mandatory, FIME can provide an optional certification process for Apple Pay Push Provisioning. Contact FIME directly for current pricing and instructions. This cost was approximately €3125 at the time of writing.

4. SDK Configuration

- **Verestro Application Ownership:** If Verestro develops the application, our development team will handle the internal configuration of the 3rd party Passkit SDK.
- **Customer Application Ownership:** If the customer owns the application, customer's development team needs to get the SDK of their choice (Apple Pay, Passkit, etc.) and follow their documentation.

Additional Notes

These instructions provide a high-level overview.

For detailed technical guidance and code implementation, please refer to the official Google Pay and Apple Pay developer documentation.

It will be necessary to implement the 3rd party **SDKs** and **APIs** provided by [Verestro](#) and used to encrypt card data and support additional processes in xPay's.

Push Provisioning & Web Push Provisioning

In this article we will focus on the process of pushing card data to X-Pays (like Apple Pay or Google Pay) – a process called Push Provisioning.

In many innovative card issuing use cases it is important to launch tokenization of cards to enable better customer experience and contactless payments with mobile. There are actually a few ways of registering card data at Apple or Google Pay from a user's perspective:

1. **Manual provisioning** – which means that the user provides card data one by one. After that the user accepts T&C, confirms this action with One Time Password received from the issuer / bank and gets the card tokenized.
2. **Scanning card** – the above mentioned process can be improved if the user can scan their payment card with their phone, so that the card data and expiry date appear automatically in the wallet. Not a difficult thing to do nowadays. This process is also very useful if a card visual can appear on the website and the user scans the card visual and data directly from their laptop.
3. **In-app Push Provisioning** – in this case, the process starts with opening a mobile banking application. The user can click the “Add to wallet” or “Register to wallet” button and can be redirected to Apple or Google Pay to finalize registrations. Various SDKs enable quick registration of the user and card data. It is a very user-friendly process.
4. **Web Push Provisioning** - in some cases it is necessary that the process of Push Provisioning starts from the website – not from the mobile app. This is called Web Push Provisioning and it is a very innovative way of starting tokenization. The whole flow is almost the same as in in-app push provisioning model. The user clicks "Add to wallet" button on the website, wallet's interface is initialized and then the user goes through the same steps as they already know from the wallet app.

This use case can be interesting in multiple projects, for example:

- **Gift cards delivery** – the user opens a gift card on the website widget on their phone and can immediately add this gift card to the wallet.
- **Expense management cards** – the user opens a card generated by their employer on their phone and can immediately tokenize the card in ApplePay.
- **Emergency cards** – the user gets a card from their insurance or bank and can use it immediately on their phone for emergency transactions.

Please contact us if you are interested in testing those use cases.

Tokenization in eCommerce Payments

Sometimes our customers ask questions connected with tokenization of eCommerce transactions. Let me focus on this topic below.

There are multiple meanings of tokenization in case of eCommerce payments.

Payment Scheme Tokenization

I think that nowadays term tokenization is the most commonly used in the context of VISA or Mastercard [card tokenization](#). Both payment schemes implemented tokenization systems (MDES and VTS) which were first used on mobile phones but today are also used for eCommerce transactions. In such situations, tokenization means that a regular Mastercard or VISA payment card gets connected to a token which is a kind of virtual card number and once this mapping happens, the user is using the token to initiate payment transactions from those cards. An example of such a use case could be Netflix which enables their users to add cards to the service and in many geographies they map cards with tokens and for all transactions a token is used rather than a regular card. In case of a security breach, a thief will steal token numbers and not card numbers.

Card on File Tokenization

Very often term tokenization is used in the context of regular **Card on File** projects. Card on File is a solution that enables merchants or acquirers to enable card payments without asking all the time for card number entry by the user. A card is held "on file" which means that it is saved in the system of merchant, acquirer, processor and can be used any time the user confirms a transaction. In such a context, tokenization can be enabled without VISA or Mastercard and tokens are owned by the processor or acquirer and shared with the merchant. In case of security breach to the merchant, real card numbers are not exposed because they are safely kept by a PCI DSS compliant processor.

Apple or Google Pay Tokenization

Sometimes tokenization in eCommerce is used in the context of Apple Pay and Google Pay transactions. In those projects the system works very similarly to the standard Payment Scheme Tokenization described above, but iOS and Android devices are used to authenticate the user and the transaction. Apple and Google enable various endpoints that merchants or acquirers can connect to and allow tokenization. Transactions are processed in a similar way to regular Mastercard or VISA transactions.

Advantages of Using Tokenization

Usually, main advantages of using tokenization are connected with security and minimising PCI DSS requirements that merchant must fulfill. PCI DSS (Payment Card Industry Data Security Standards) require that each entity, which holds and processes card data, performs a set of actions to make sure that cards are securely processed. Very often tokenization brings additional benefits like cost optimisation and user experience improvements.

However, once you think about such a project, take seriously all potential long term impacts. If you do a project directly with your acquiring institution you may get seriously impacted in case you want to change provider. The last thing you want as an eCommerce merchant is to have a monopoly of your payment provider. You should choose providers that work with multiple acquirers, can provide tokenization or card on file solutions for users cards and tokens without being limited to a single acquirer. In such cases you can get acquiring offers from multiple acquiring partners and enable competition to lower prices of payment processing.

Please contact us if you consider such projects.

Web Push Provisioning - Overview

With Web Push Provisioning, users can add a card to Apple Pay or Google Pay directly from a browser. For Issuer or Fintech it means that the user doesn't need to have your app installed on their device — they can use your web application instead. Tokens created in this way are saved in cloud and are assigned to the user's profile instead of a device. This means that the card is accessible from any given device - cardholder may even use Google Pay on their iPhone! This article explains how it works and what it takes to build it with Verestro.

How tokenization works

When a card is added to a digital wallet, what actually gets stored there is not the card number itself, but a network-level token that represents it. This process — tokenization — means the card is ready to use immediately after issuance, both in-store and online, without exposing sensitive card data at any point in the transaction.

With Web Push Provisioning specifically, the token is cloud-based and tied to the user's wallet account rather than a specific device. This is what makes cross-device access possible: the same card can appear and be used across all devices connected to that account.

Web Push Provisioning vs. In-app Push Provisioning

Web Push Provisioning works in much the same way as in-app provisioning — the key difference is simply where the flow is initiated. Instead of a mobile application, the user starts and goes through the whole process from a web browser. Everything else, including the tokenization process and the wallet confirmation step, remains very much the same. User clicks "Add to wallet" button, sees the screens looking similar to the ones they know from wallet mobile apps: device selection, terms & conditions, address confirmation (for Google), OTP step up (if procced) and confirmation screen. Also, integration and the whole process logic are similar in both flows.

How to implement Web Push Provisioning with Verestro

TMP API — The core of the integration. Your backend calls the Verestro TMP API to initiate the provisioning request and retrieve the encrypted parameters needed to complete the wallet flow.

Endpoint to be called is [signed-cards](#) and documentation provides examples on how the request for each use case should look like.

Wallet JavaScript SDKs — On the frontend, you implement the Apple Pay JS or Google Pay API library. These handle the wallet confirmation UI natively on the user's device and pass the provisioning result back to your page. It's also possible to have whole frontend initialization implemented by Verestro.

Keys & certification — At the beginning of the project you will need to onboard with Google and Apple. Both Apple and Google require certification before you can go live. For Apple you will need a certified Lab, for Google there is a process called self-certification via their developer platform. Verestro advises you through the whole process, starting from acquiring the PGP keys from Google up to finishing the certification with Apple.