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.

Revision #3
Created 19 May 2026 13:49:25 by Ignacy Korytko
Updated 19 May 2026 14:21:39 by Ignacy Korytko