Progressive Web Applications – killing two birds with one stone

Web applications launched like normal websites, but working like native mobile applications will be a revolution in the business world. They will save their creators time – and effort.

The phone screen is reading a fingerprint. It checks out. "The customer asks to update the offer..." – a customer advisor hears, driving their car. The status comes from their PWA application and is related to a push notification about an upcoming business meeting.

Although fingerprint login and push notification are native functions, the application the advisor uses is made available by their bank on an internal website, and launched from an icon on the home screen of a mobile phone.

It launches and works like a native application. But a native application it is not.

Steve's vision

Few people remember that back in 2007, Steve Jobs and Apple saw the future in web applications, not native ones, i.e. those developed for a specific platform: Android or iOS.

And then came the AppStore. But even then, on the iPhone, you could clip a web app to your home screen and run it directly from an icon – doing what Progressive Web Applications do today.

These days, the temptation to build them is huge. Why create something twice, for one operating system first, then for another next, when you can get it right in one swing? All the while using popular technologies for building web applications...

Pros and cons

There is a demand for web applications on the market. They are successfully used by banks – in the form of hybrid applications. This is because, from the banking point of view, native solutions have a number of cons, which are:

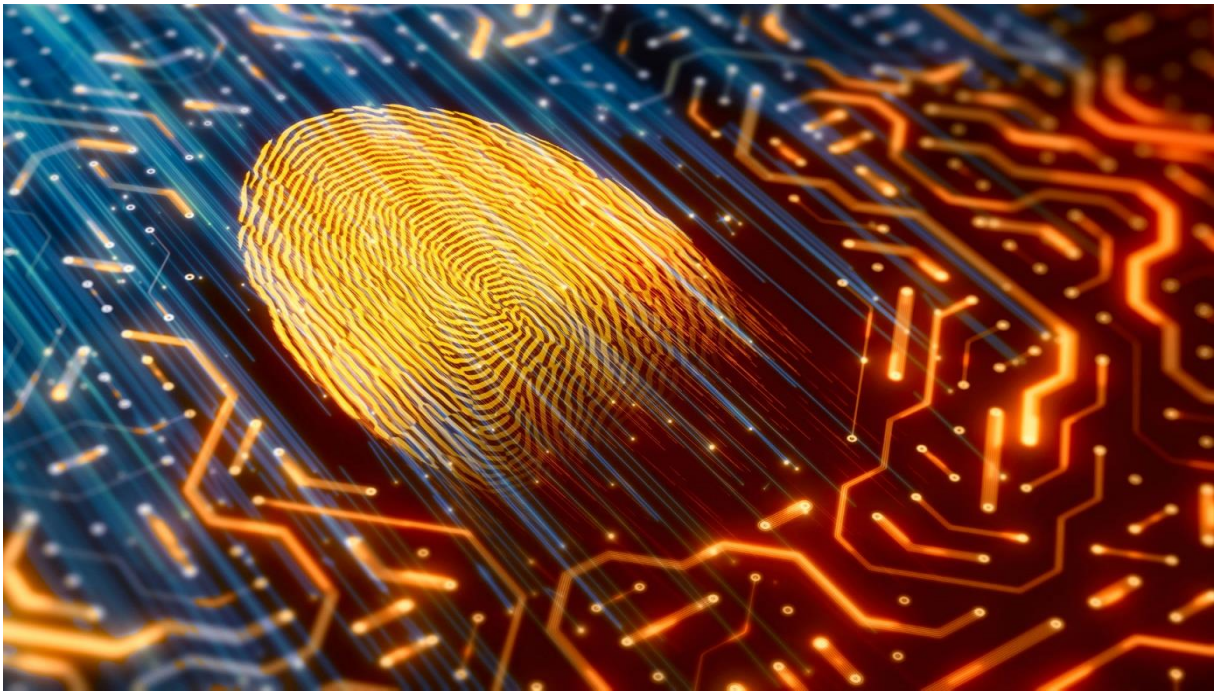
- duplication of competences

- duplication of functions and the need to maintain them

- the need to adapt all applications to legal regulations

- the need to synchronize changes to ensure consistency

These problems do not apply to hybrid and PWA applications. PWAs also have a number of additional pros. They can be made available via links. They don’t need to be published in app stores – this eliminates long delivery times for new features and bug fixes. There is also no risk that the publication itself will take a long time because a store will reject it until fixes are made.



Threat or opportunity?

You might think that PWAs are a threat to app store owners. But as SensorTower data shows, for Apple and Google PWAs account for about 20% of revenue. Moreover, in 2019, 68% of AppStore revenue came from games. For Google Play, games accounted for 84% of revenue. This means the lack of business applications in stores is not a real threat to online giants.

Still, Apple is chasing the market when it comes to PWA. An example can be the lack of fingerprint support in the iOS system. This is due to the fact that PWAs do not use native device functions in the form of extensions – as hybrids do – but work based on current browser capabilities.

It still happens that stores are reluctant toward hybrid applications that do not use native device functions. There is a stereotype of a hybrid application being a workaround; a "frame" for browser content. PWAs change this thinking, and, at the same time, create a space for web applications for mobile devices.

One step to PWA

The basis for PWA is Responsive Web Design (RWD), a technique for designing a website so that its appearance and layout automatically adjusts to the size of the browser window.

Our experience from working on the Comarch Corporate Banking system (online banking for corporations and SMEs) has shown that the cost of RWD is just several percent of the total software development cost. This is relatively little in relation to the cost of developing two native applications.

And since there is only one step from RWD to PWA, even today you can find many successful implementations of PWAs by such global brands as AliExpress, Alibaba, Flipkart, Twitter, or Trivago.

Banking comes next?

This shows that the time of PWA in banking will come. The first use case could be mobile applications designed for bank employees - e.g. for business client advisors. Such applications would improve their work: they would organize information about meetings with clients, analyze clients' needs, or help maintain client relations.

Other use cases are applications designed for bank customers. Currently, most banks have introduced RWD in their applications for online banking. One of the reasons is that usually not all features are available in native versions of such applications. In these versions it’s not possible e.g. to change card limits, negotiate exchange rates or delegate privileges to another employee.

From the banking point of view, it may be cost-effective and operationally reasonable to provide niche functions and specific products as RWD solutions, especially to companies. However, the market is yet to see this coming.

Maciej Bachmiński, Product Owner, Comarch