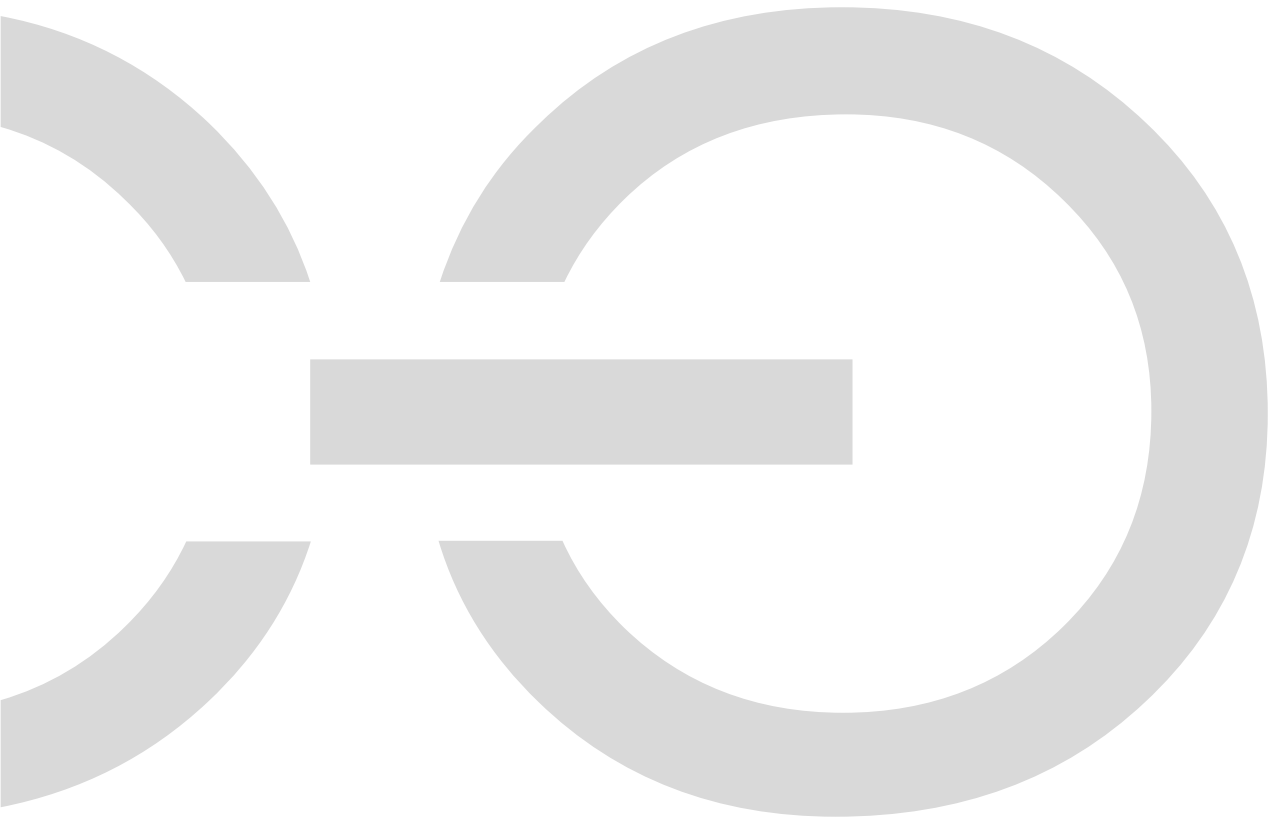


Architecture of WEBCON BPS mobile application



1.	Application assumptions	3
2.	Hybrid application	3
3.	Architecture of the mobile application	5
1.1.	Diagram.....	5
1.2.	Authentication.....	5
1.3.	Mobile application	6
1.4.	Webview – WEBCON BPS Portal	6
1.5.	REST API for the mobile application.....	6
1.6.	Cordova	6
4.	Configuration in WEBCON Designer Studio	6
5.	Architecture differences for version from 2019.1.1 to 2020.1.2.141	7
1.1.	Functionalities division between native part and Webview	7
1.2.	REST API for the mobile application	7
1.3.	Cordova	8

1. Application assumptions

WEBCON BPS 2020 mobile application facilitates working with WEBCON BPS system from the level of mobile devices.

Basic assumptions of the application architecture:

- Dedicated application for mobile devices
 - Android – support from version 4.2
 - iOS – support for the last two versions
- Works in two modes depending on the WEBCON BPS system version:
 - From version 2020.1.2.163 and higher – the new mode is based on the WEBCON BPS Portal embedded view, the architecture is described in 1-4 paragraphs
 - From version 2019.1.1.x. to version 2020.1.2.141 – the mode is based on the native implementation of the application lists, processes, and user's tasks. Differences in architecture with regard to the new mode is described in 5 paragraph
- Hybrid application
- Application works online – requires the internet connection with WEBCON BPS installation
- Application works based on the users of connected WEBCON BPS system

Applications for individual platforms are available in application stores (suitable for the operating system of the device) dedicated to these platforms.

When it comes to the Android system devices it is the option to provide an installation package to the client (.apk file) which allows the application to be installed on devices that do not have access to Google Play store.

2. Hybrid application

WEBCON BPS 2020 mobile application has been implemented as a hybrid application. Some of the application functionality is implemented in a classic form – as the native mobile application, while the vast majority of the application is embedded by a browser window docked in the application (called Webview). WEBCON BPS Portal web application view has been embedded in a Webview, so most WEBCON BPS functionalities is supported in the same way as in desktop browsers. The Webview is responsible for user authentication.

Native part

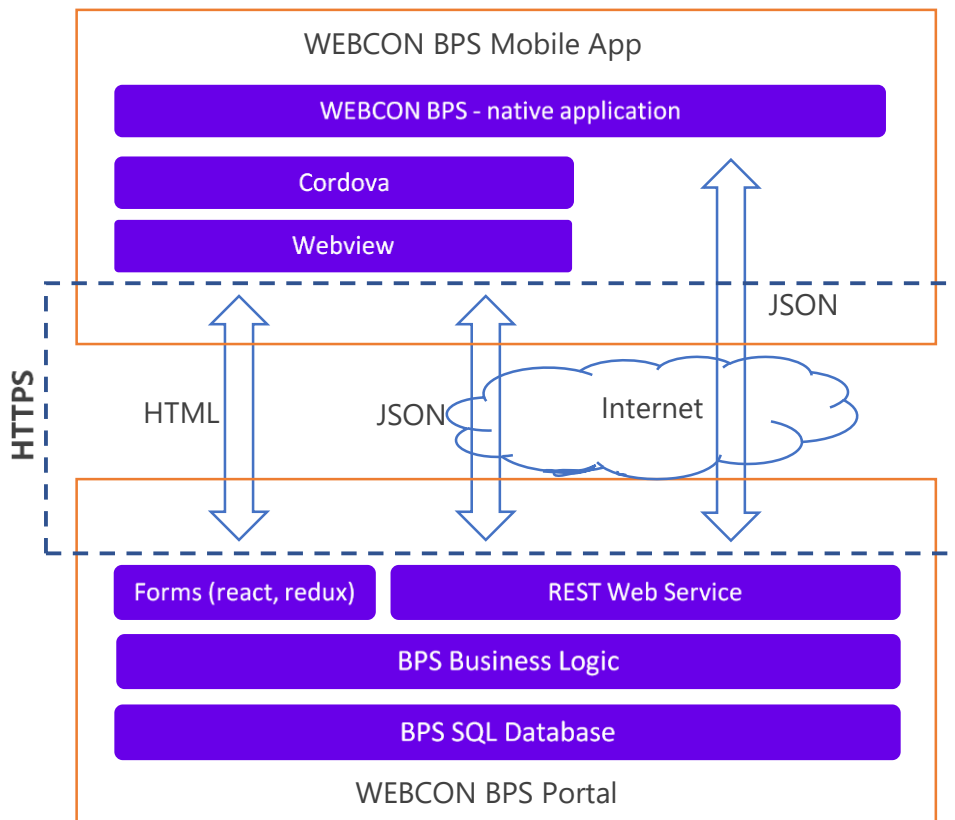
- Profiles, logins and PIN security support
- Main panel of the application with navigation buttons in Webview
- Recording a voice message – Cordova
- Log history
- Barcode/2D codes search (only for iOS)
- PUSH notifications (only for iOS)

Webview

- Authentication
- WEBCON BPS Portal

3. Architecture of the mobile application

1.1. Diagram



1.2. Authentication

Authentication in the mobile application is performed using the access mechanism and tokens refresh.

Using the WEBCON BPS Portal application view facilitates logging in the mobile application, and you can authenticate yourself by using all active login providers (including ADFS, AAD, Webcon BPS Auth) and it is similar to a standard browser.

After selecting a provider, a user is redirected to the login page, from where, after entering the correct credentials, they will return to WEBCON BPS Portal handing over the generated access token to the mobile application.

The access token is used in all REST API invoke methods and when entering on the Portal's main page (a token is added to the request header):

- The access token lifetime is 1 day; when it expires – it is refreshed using the refresh token
- The refresh token lifetime is 26 days; when it expires – you have to authenticate again

Azure proxy enforces a different method of authentication and it is currently not supported by the mobile application. Work is underway to service it.

1.3. Mobile application

WEBCON BPS mobile application works only in Online mode – all data presented to a user and data entered by a user are downloaded/transferred to the connected instance of WEBCON BPS system installation on an ongoing basis.

The communication is performed by the native part of the application which invokes REST API services available the WEBCON BPS installation.

Data using in this communication is transferred in the JSON format – this applies to both invokes to WEBCON BPS and responses to the mobile application.

1.4. Webview – WEBCON BPS Portal

In the Webview, a user has access to the full interface functionality of WEBCON BPS Portal.

For more information about this functionality see: <https://howto.webcon.com/webcon-bps-2020-mobile-application/>.

1.5. REST API for the mobile application

Communication between the mobile application and WEBCON BPS is performed via REST API invokes and embedded Portal.

The most part is supported by WEBCON BPS Portal, hence Webview of the mobile application requires access to the same html pages and REST services as the application running in the browser.

In addition, the mobile application uses REST API dedicated to the following functionalities:

- Downloading version of the application
- Downloading the configuration
- Device registration and downloading the security level
- Downloading the PUSH configuration (full support only in iOS)

All dedicated REST API services for mobile application are available at:

<https://<BPS Portal>/api/mobile>

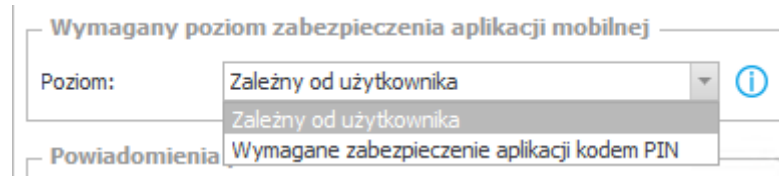
1.6. Cordova

The mobile application uses the Cordova framework as an interface between the native part of the application and Webview. Cordova is currently used to record voice messages. Each recorded message is saved on the mobile devices in the WEBCON BPS 2020 folder.

4. Configuration in WEBCON Designer Studio

The WEBCON BPS mobile application requires:

- Process configuration -> Form view -> Compact form
 - The settings of the compact form are inherited by default from the basic form. There is the option to break this inheritance and separate configuration for the compact form.
 - In the same way you, can manage the compact form behavior using the form rules. By default, they are inherited from the basic form, but you can break the inheritance.
- System configuration -> Global parameters -> Required mobile application security level



This setting is common to all WEBCON BPS processes and applications. It allows you to enable the enforcement of entering the PIN code each time when WEBCON BPS mobile application is opened, regardless of the security level selected by a user. When on the mobile device, the Touch Id is active – it will be used. If login is incorrect, a user must enter the PIN provided in the configuration of the mobile application.

5. Architecture differences for version from 2019.1.1 to 2020.1.2.141

The general architecture of the mobile application described in the previous chapters is common to both application operation modes. The differences are described below.

1.1. Functionalities division between native part and Webview

Most of the functionality is implemented on the native part of the application code. The division of the application elements is as follows:

Native part:

- Profiles, logins and PIN security support
- Main panel of the application
- Navigating by report views
- Searching
- Starting new workflows
- Application, processes and instances lists preview
- Log history
- Searching instances by barcodes/2D codes (only iOS)
- PUSH notifications (only iOS)

Webview

- Authentication
- Forms of the preview, edit and adding new instances
- Graphical reports

1.2. REST API for the mobile application

This mode uses the same dedicated REST API services for the mobile application. Below there is the extended list of the functionalities which use REST API:

- Downloading the version of the application
- Downloading the configuration
- Device registration and downloading the security level
- Downloading the PUSH configuration (full support only for iOS)
- Downloading the assigned task counters

- Downloading the list of applications, processes and instances
- Downloading data for report views
- Searching

1.3. Cordova

Adding attachments support from the form level works based on the Cordova interface. It allows you to add voice messages, images directly from the device and other files from the device memory.