BPA Solutions

SPFx App Technical Documentation

BPA Solutions

October 2024

BPA Solutions – US : +1-888- 556-9262 - Switzerland : +41 24 524 25 40 – http://BPA.Solutions

Contents

1. BPA Applications on Office 365/SharePoint 3
2. Architecture Overview 3
Overview 3
Infrastructure Architecture 4
Authentication & Authorization 5
2nd Entra ID app approach for permissions (optional) 6
PnP PowerShell approach 6
Client Data Security 7
Main application components 7
Azure Environment Security 8
3. Required Permissions 8
Client side (SPFx) permissions 8
Entra ID SharePoint and Graph WebAPI permissions 10
4. BPA Licensing System 11
5. BPA SPFx Strategy 12

1. BPA Applications on Office 365/SharePoint

BPA Apps (Quality, Medical, CRM, App Builder) are SharePoint apps to be installed in the client Office 365 environment and developed with the SharePoint Framework (SPFx) technology.

2. Architecture Overview

Overview

BPA SPFx applications are divided in 2 subcomponents:

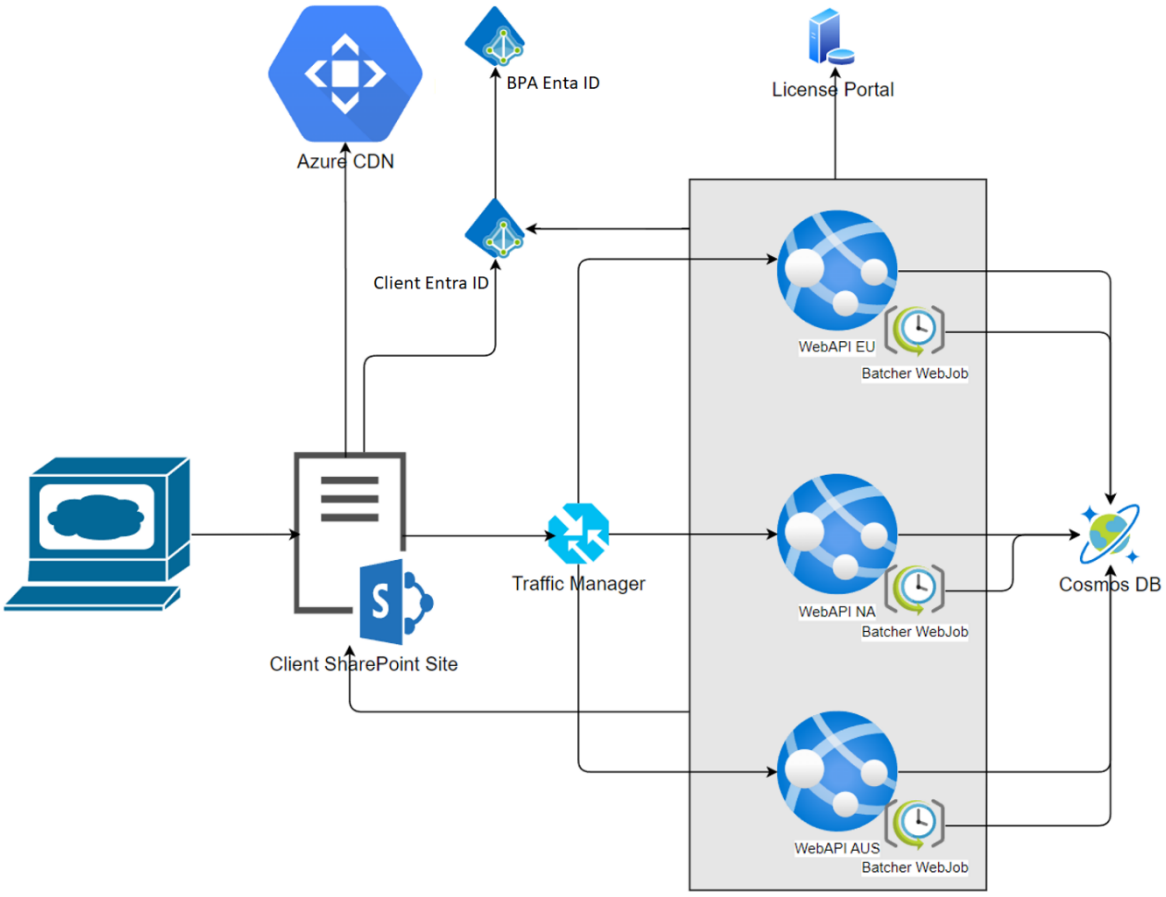
- SPFx webparts, interacting with users, working with the permission token given by the current user session. Webparts are installed by the app in the client environment. SPFx active part is hosted in CDNs and only a non-changing part hosted in App Catalog. So, updates (usually monthly) can be done without to update all clients' sites.
- A multi-tenant WebAPI called by the SPFx client and interacting with SharePoint, using:
    - the user token for operations to execute with delegated permissions.
    - the app token obtained thru Microsoft Entra ID (Entra ID) certificate for operations the user isn't allowed to execute.

The WebAPI runs in a Microsoft Azure environment maintained by BPA. Software updates are periodically done by BPA on the Azure backend with no impact on clients data and app configurations.

The WebAPI is used for:

- user actions with high level permissions
- remote event receivers like field replications when an item is created/updated
- batch actions
- complex & secure operations
- eSignature

Infrastructure Architecture



BPA infrastructure architecture

Components:

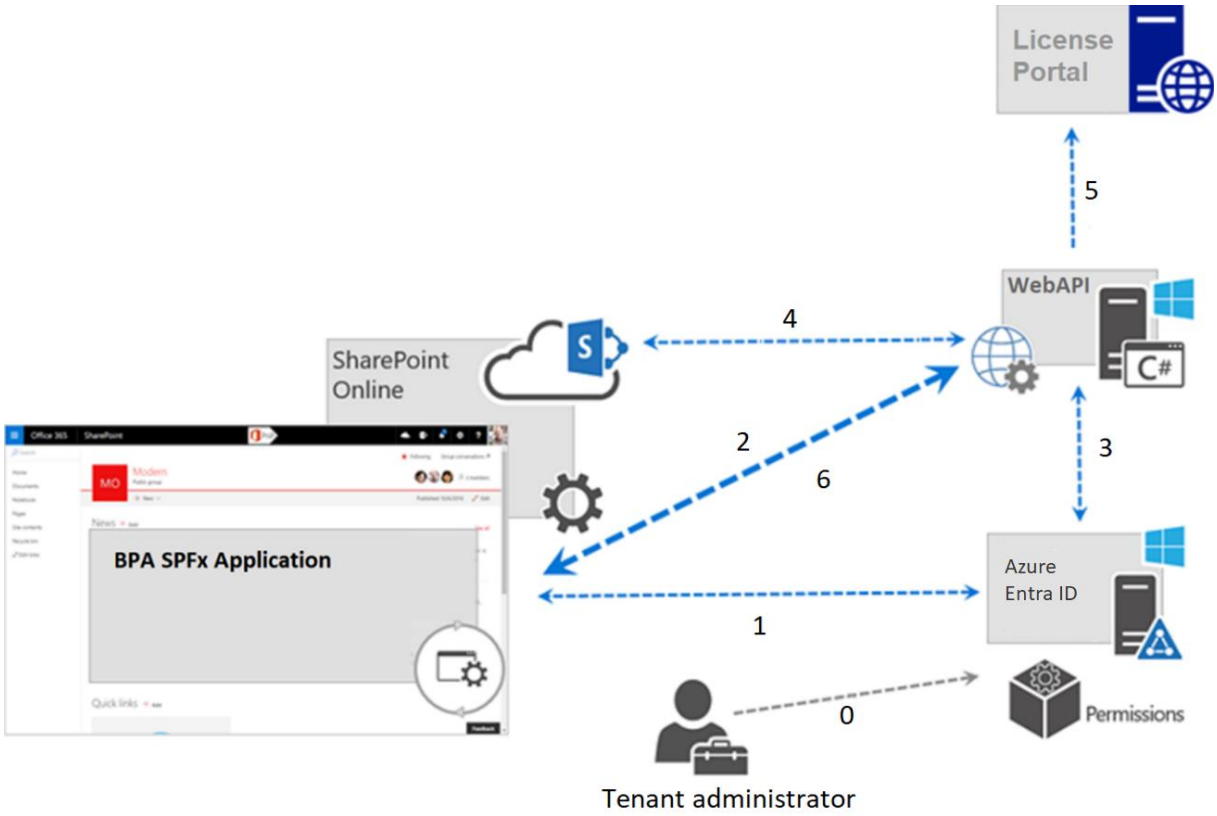
BPA hosted:

- Azure CDN – serves JS files for SPFx web parts. Provides stability and improves performance. We store it separately minimize the need for package updates for clients. For dedicated clients, we store JS inside SPFx package (SharePoint Online CDN) to reduce installation complexity.
- Traffic manager – redirects WebAPI requests to the closest WebAPI server.
- Location based app services – WebAPI backends to process client web parts requests.
- Location based batcher web jobs – timer jobs to process on a schedule.
- Cosmos DB – a cloud DB used for configuration (like mapping between batchers and sites to process) and distributed cache of client site structure.
- License Portal – proprietary portal to manage client license and licensed users.
- BPA Azure AD – stores the application registration with required permission definitions. Changes in requested permission here does not apply to clients until consent is (re)granted by client tenant administrator.

Client hosted:

- Client Azure AD – issues authorization tokens for both client web parts and WebAPI/WebJobs. Also contains the enterprise application with granted permissions. It's possible to revoke permission after granting it during initial consent grant to the whole Entra ID application. In case when some permissions are not planned to be used. Like sending email using Graph for example.
- Client SharePoint Site – hosts client web parts on pages and provides Graph & SharePoint APIs to access data

Authentication & Authorization



BPA SPFx Apps authentication and authorization flow.

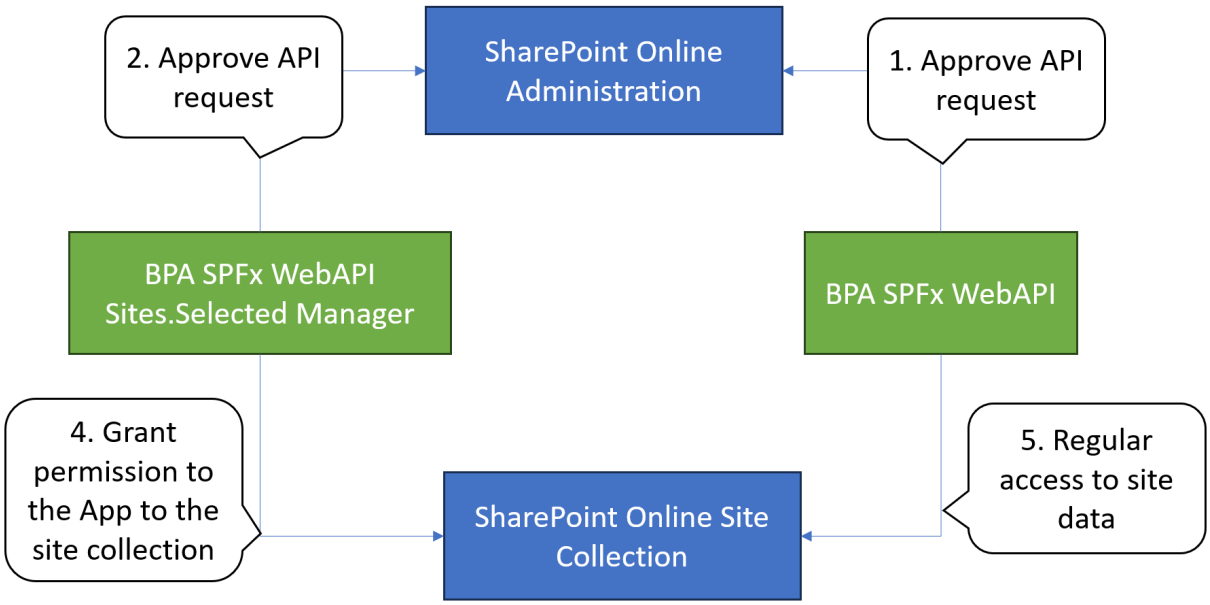
0. Tenant admin grants consent to **BPA SPFx WebAPI** Enterprise application & API requests
1. JS web part securely gets access token to the WebAPI using **user_impersonation** scope
2. JS web part sends request to the WebAPI using the received token specifying the current site url using HTTPS protocol.
3. WebAPI validates the user token and gets access token for either user or app only based on the currently executed method logic.
4. WebAPI access SharePoint APIs with the received access token
5. WebAPI checks user license in our License Portal system.
6. Response returned.

All communications between the systems are done under HTTPS protocol (WebAPI, License Portal, SharePoint Online, Entra ID). BPA application architecture was validated by a Microsoft MVP expert.

2nd Entra ID app approach for permissions (optional)

To simplify the installation for non-technical users, we decided to use an additional app (**BPA SPFx WebAPI Sites.Selected Manager**) to grant Sites.Selected FullControl application permission automatically during the installation to the main **BPA SPFx WebAPI** app to the site collection app is being installed to. Using this approach, we follow the least privileges principle, as 2nd app is optional (possible to use PnP PowerShell commands to do it) and after installations 2nd app can be fully removed from the tenant.

NOTE: PnP PowerShell commands use the same approach - Authentication | PnP PowerShell but requires additional permissions outside of SharePoint.



1. Approve API request for **BPA SPFx WebAPI** and grant contest to Sites.Selected application permission in general (without specification of the exact site collection(s)) for regular usage of Graph/REST/CSOM APIs.
2. Approve API request for **BPA SPFx WebAPI Sites.Selected Manager** and grant contest to Sites.FullControl.All delegated permission to be able to grant Sites.Selected FullControl.
3. FirstLoad web part will check permissions to access the current site and if not, it'll try to automatically grant them using BPA SPFX **BPA SPFx WebAPI Sites.Selected Manager**. If it fails (API request not apptoved), it'll show a message how to grant them using PnP PowerShell.
4. Main (BPA SPFx Web API) app can access the site.

PnP PowerShell approach

```
$siteUrl="https://you-tenant-name.sharepoint.com/sites/bpa-site"
$appId="BPA_SPFx_WebAPI_App_ID"
Connect-PnPOnline -Url $siteUrl -Interactive
Grant-PnPAzureADAppSitePermission -Permissions "FullControl" -Site $siteUrl -AppId $appId -
DisplayName "BPA Sites Selected"
```

We need this API request to be approved to automatically set Sites.Selected permission on the current site collection for BPA SPFx WebAPI to follow the least privileges approach. If you do not want to approve it, you have to grant manually the permissions using the listed PnP PowerShell commands on your print screen. We suggest approving this API request, as it makes it easier for end users. After the installation, the API request and Entra Service Principal/Enterprise application can be removed. But if you try on a new site after removing, it'll lead to the same screen asking to do it manually. So, there is a choice for end users/admins to leave it for new sites (it does not affect security as it's delegated permission) or remove and do it manually with PnP PowerShell.

BPA SPFx WebAPI Sites.Selected Manager – secured access to 2nd helper application to smoothly grant Sites.Selected application permission (we use least privileges approach, thus asking Sites.Selected application permission only, instead of much higher Sites.FullControl.All) for **BPA SPFx WebAPI** to the site collection with the BPA components installed. This API request can be removed (together with Service Principal/Enterprise Application) after successful installation as needed only during the installation. There is a guide how to do it on the support site - https://www.bpa-solutions.net/wp-content/uploads/2023/11/BPA_Support_BPA365_ModernApps_removal_guide_SPFx_WebAPI_Sites_Selected-Manager.pdf. Based on our experience with Classic Apps, many AppSource users are not technical specialists, thus it's very difficult for them to use PowerShell to install it with proper version, load PnP modules and grant permissions. But for the security concerned administrators, it's possible to reject this API request and grant permissions manually. First Load web part will generate PowerShell commands to execute if it failed to grant permissions automatically and will have not access to the site collection using **BPA SPFx WebAPI**.

Client Data Security

Data is stored in regular Microsoft SharePoint lists and libraries, in the client Office 365 environment. **BPA doesn't host any client data**. SharePoint security and protection apply and are fully respected.

All SharePoint features like versioning, recycle bin and backup/restore are fully available.

No client data is stored outside of SharePoint:

- No client data is stored in the BPA Azure app service (WebAPI & Batcher (Timer WebJob)).
- No client data is stored in the BPA SPFx application.
- No user password is stored anywhere.

SharePoint and application permissions are under control of the client tenant administrator.

If requested, the client can host his own WebAPI server inside his own Azure tenant. In this case, the client is responsible to manage BPA software updates.

Main application components

BPA Application includes a variety of components. They change in each version with the new components or features. Here are 3 major parts of application with some inner components:

- SPFx client side:
    - SPFx Webparts to CRWD data from SP lists. Current webparts are BPA MasterDetail, BPA DataViewer, BPA Form, BPA MailMerge, BPA Charts, BPA eSignature and some BPA settings functions.
    - SPFx Extensions (e.g. to display the BPA Navigation)

    - First Load web part to guide user through the initial configuration.
    - All client web parts calls to SharePoint are done only with the current user permissions.
- WEBAPI side:
    - Serving information to itself and to the SPFx client about the site structure - list the lists, their content types, views and fields.
    - Processing SharePoint Online Remote Event Receivers (RER) (like OnUpdated event set up by BPA Reminder or BPA Fields Replication) and CRWD data from SP lists according to the task to achieve.
    - Depending on the function to achieve, the action can be run with current user permission (exporting list of items to Excel that user has access to) or with the App level permissions (e.g. RER or batches).
    - eSignature for secure signing of items or documents with possible invalidation using RER
- Batcher side:
    - Running batches, e.g. for Reminders or MailMerge.

All data transfers are done under HTTPS protocol.

Azure Environment Security

The WebAPI runs in a Microsoft Azure environment maintained by BPA. This environment is protected by MFA for restricted administrators with restricted port access. State of the art security systems are in place. Backups are periodically done and prepared for high availability in case of issues.

3. Required Permissions

The app needs the described below permissions to work normally, which is fully standard for such an add-in. Installation must be performed by a Tenant Global Administrator to be able to grant all requested permissions.

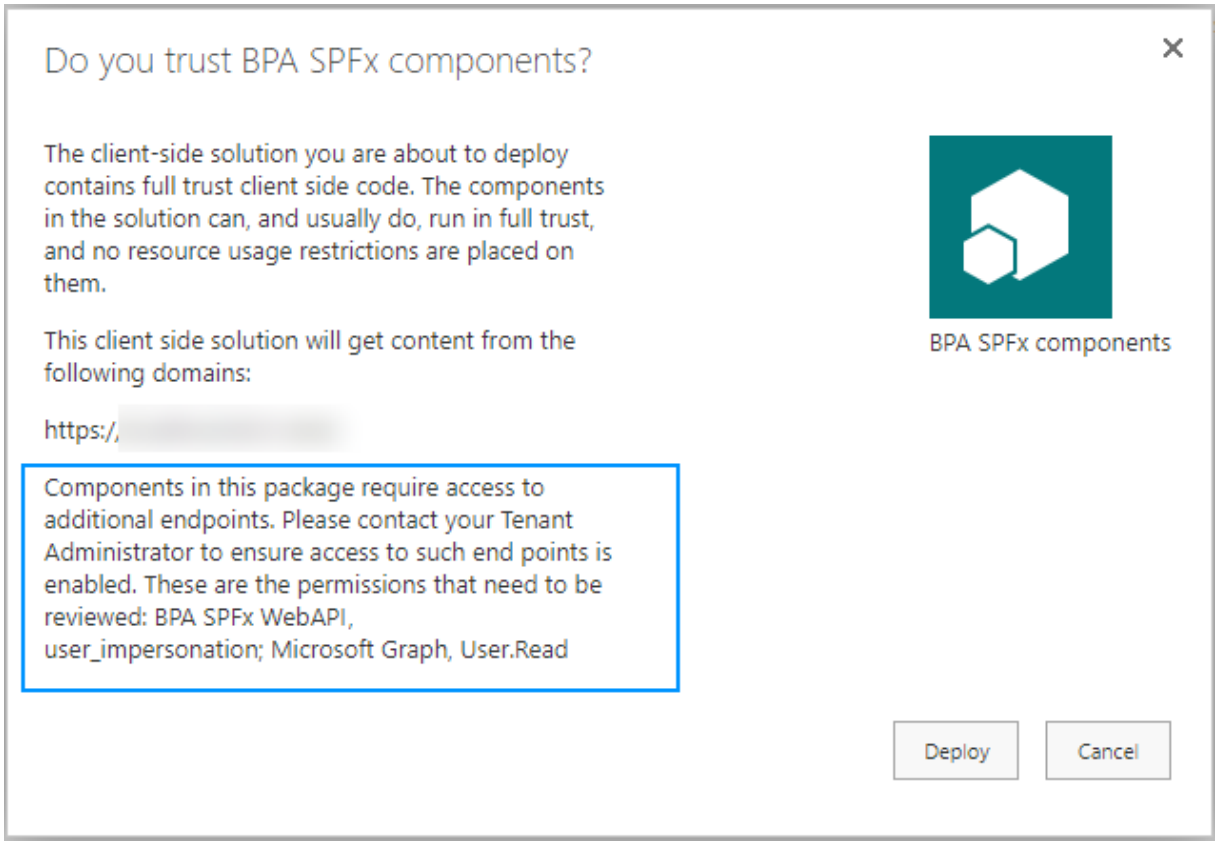
In addition, it requires these features to be activated:

- (optional) Custom Scripts have to be activated for the BPA Site Collection.
- App-Only access has to be allowed on tenant. This requires SharePoint Online Management Shell.
- Granting Sites.Selected permissions to the site collection BPA applications will be installed with either FullControl or Manage level.

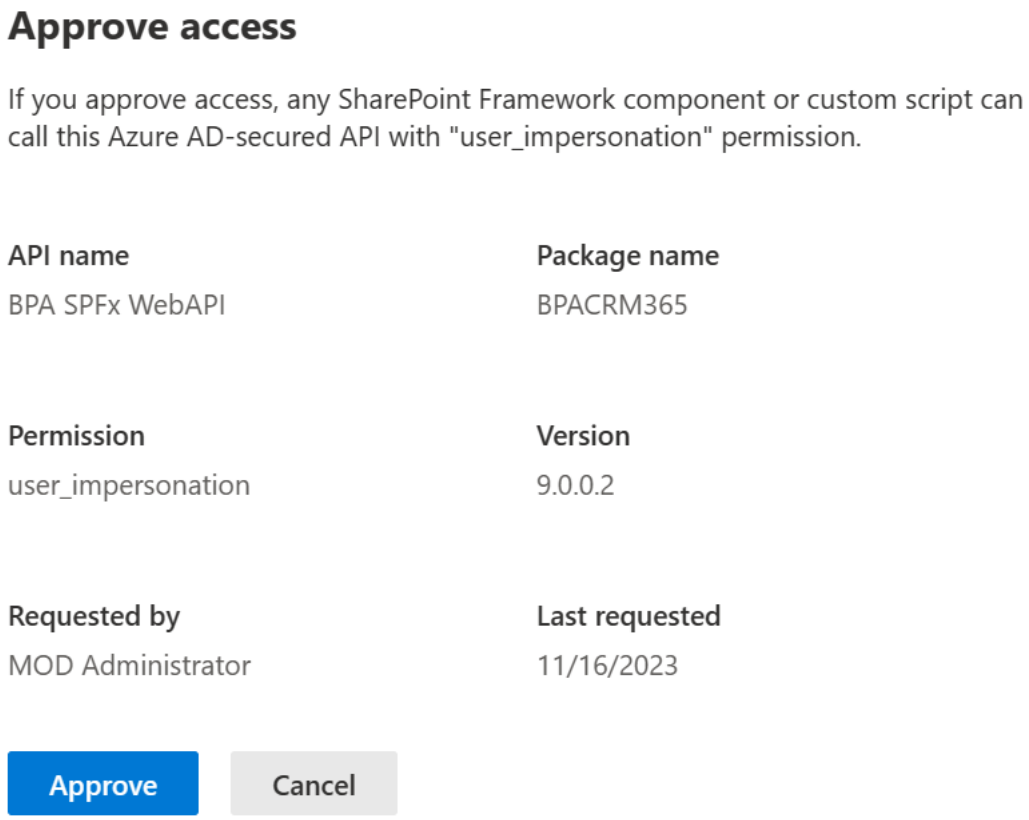
Client side (SPFx) permissions

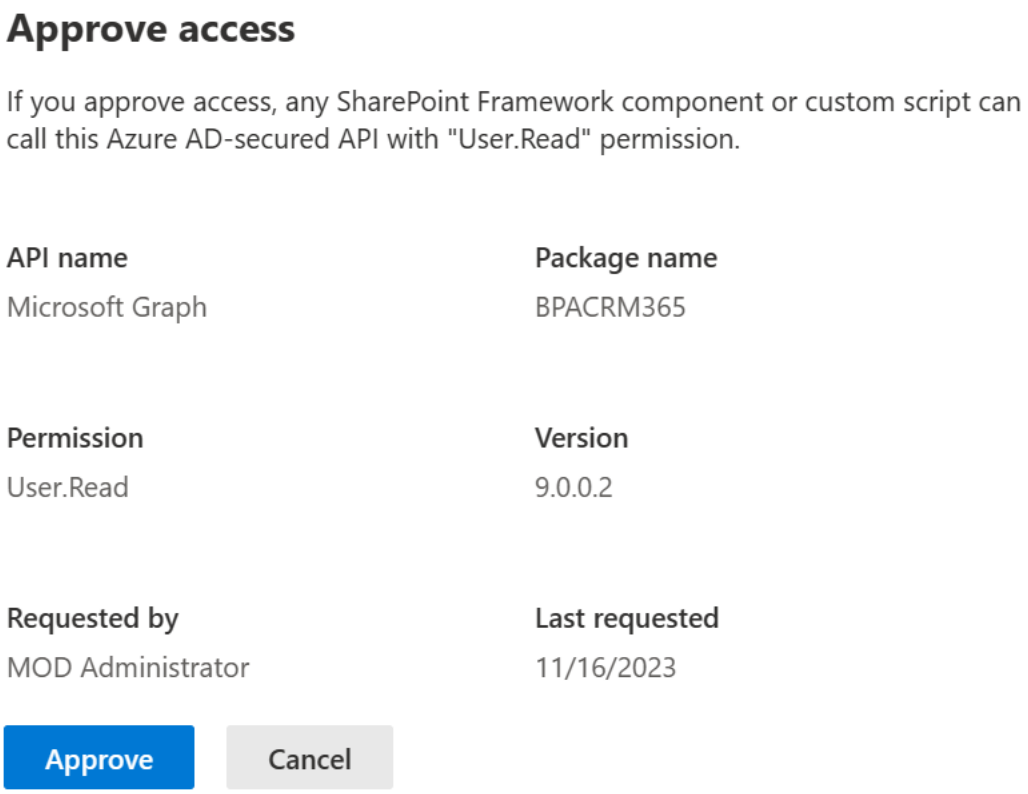
The application requires these permissions to be granted for client-side part to be able to access Entra ID protected resources from JS code:

- **user_impersonation** – Delegated - to protect WebAPI backend with Azure AD (allow calls from only authorized users)
- **User.Read** - Delegated - get basic information about logged in user on client side



These permission requests need to be approved in SharePoint Admin Center:

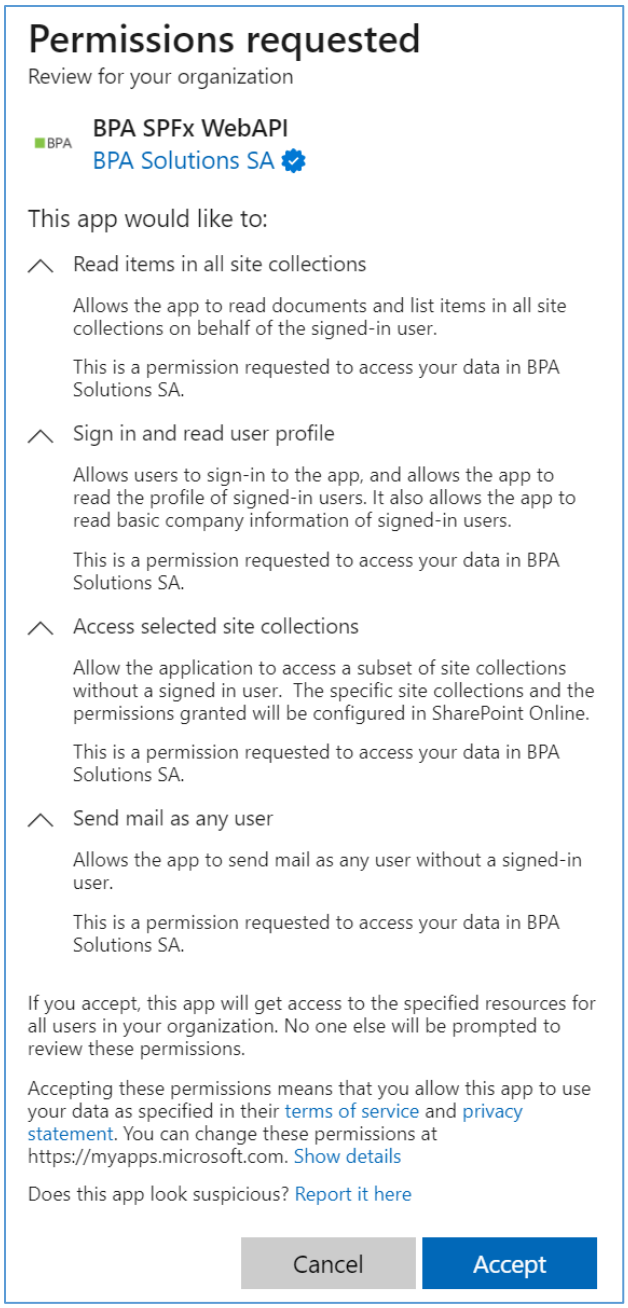




Entra ID SharePoint and Graph WebAPI permissions

The application requires permission to Graph and SharePoint using Entra ID authentication for both app only and user to perform actions on server side:

- **Mail.Send** – Application - send email notification from the configured user using Graph API
- **Sites.Selected** – Application - have access to the ONLY site collection using Graph API (i.e. BPA Mail Merge convertion to PDF)
- **Sites.Selected –** Application – to have access to the ONLY site collection using SharePoint APIs – CSOM and REST
- **User.Read** – Delegated - get basic information about logged in user in WebAPI
- **AllSites.Read** – Delegated – Required for Export to Excel feature to read all items user has access to in a list/library in WebAPI using CSOM API.



Read the app installation procedure for more information about granting these permissions.

4. BPA Licensing System

The license portal is the most important part of the BPA licensing system. It's a BPA-hosted SharePoint site including the following information: the number of licensed users and readers, site URL(s), and the licensed users/readers accounts. The license portal does not store any sensitive user data – only user accounts (usually email address) and site URL(s) are stored. No user password is stored or used by the license portal. The license portal requires a named license manager(s) to get an access to the license portal with a password, and receive notifications. The licensing system features a scheduled job to process expiration notifications and a web service to securely communicate with the WebAPI to check

attributed licenses and manage licensed users (this part is protected by an auto-generated license password that is shared with the client license manager and is required to view/change licensed users).

There are 2 user roles in the licensing system:

- Reader – can only view items/documents
- Full user – can fully use the system

The BPA licensing system does not check user's permissions in SharePoint to define the current user role. The user license is automatically attributed at the first user connection to the site with the BPA App, based on the default user mode set in the licensing system. Clients can change the user mode and remove inactive user licenses directly from the licensing system in the BPA App settings, with no need to ask BPA. Note that the user role does not limit user actions in BPA apps (but that might be introduced later).

The licensing system verifies the license when each user connects to the site. Information is cached for 24 hours to keep good performance and reduce the number of server calls. It's possible to do a direct call and recheck the user license in case of an issue.

During the license verification, the following information is securely sent from the WebAPI to the License Portal web service:

- Site URL
- User account
- Product

The licensing system allows to register for a free 30-days trial, or to extend/change the license details.

5. BPA SPFx Strategy

BPA SPFx Apps is our last generation development using latest Microsoft development technologies, like Microsoft SPFx, PnP and React, and succeeding to previous on premises/cloud hybrid apps.

BPA's SPFx strategy is the following:

- The security of the application is ensured by all Microsoft Identity Platform for Microsoft SharePoint and Microsoft SPFx framework. We conform to all Microsoft recommendations and our decisions around data access were validated by experts.
- We are using a unique architecture: SPFx/PH for 365/online.
- SPFx Apps only supports SP365 modern pages.
- SharePoint access is done with the CSOM/REST APIs.
- It works with the current user permissions (JavaScript/React, on the client side) or with user or App permissions (C#, on the PH side).
- SharePoint lists, content types, views and fields caching is optimized (PH side).
- We store app configurations in hidden SP lists with unique identifier, sites synchronization capabilities and readability from C# and React/JavaScript.
- Our reliable business components are optimized for CSOM.
- Keeping the best App performance is our main goal.