



Dynamics 365 & Power Platform UK User Groups



Samir Daoudi
02/11/2022



**MICROSOFT TEAMS
INTEGRATION WITH
POWER PLATFORM**



Samir Daoudi

- 18 Microsoft Certifications, Power Platform, SharePoint, M365, SQL Server & .NET.
- 12 Years' Experience with Microsoft Technologies Stack.
- Blogger and Technology enthusiast.



<https://www.daoudisamir.com>



[@daoudi_samir](https://twitter.com/daoudi_samir)



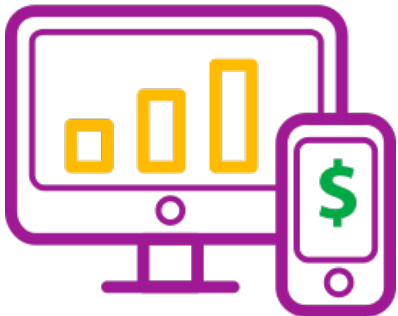
<https://www.linkedin.com/in/samir-daoudi-92b19330/>





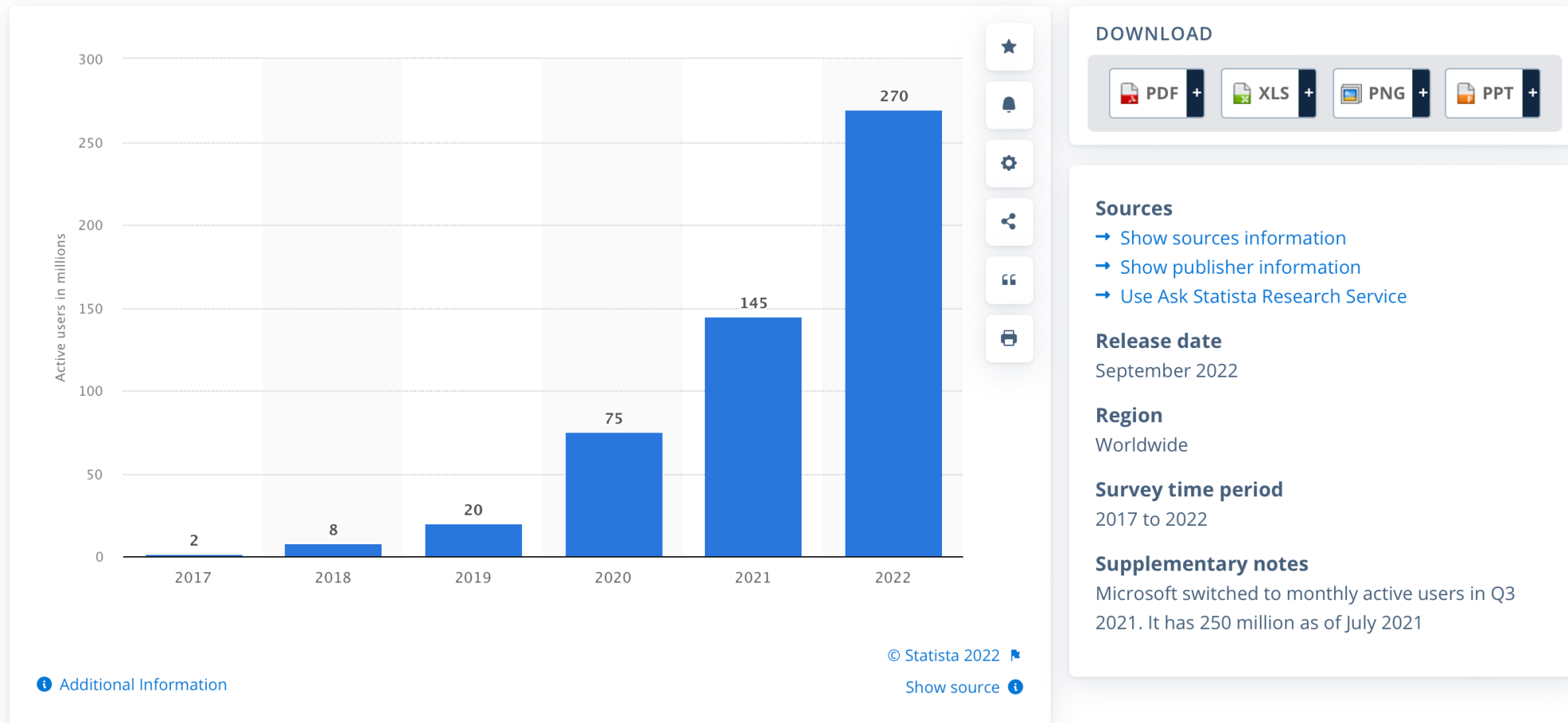
Why Building Applications for MS Teams?

- Since the start of the pandemic, users are spending most of their time in Teams, and building applications For or Within Teams is becoming a trend.
- It avoids context switching and improves productivity by reaching to your users where they spend most of their time, i.e. Teams.
- Easy to deploy, especially with the increase of Personal devices usages, where IT teams have less control over what applications get deployed to the users.
- Apps published and surfaced in Teams will show to the users regardless which device they use.



Number of daily active users (DAU) of Microsoft Teams worldwide as of 2022

(in millions)



<https://www.statista.com/statistics/1033742/worldwide-microsoft-teams-daily-and-monthly-users/>



Different methods of Publishing PowerApps within Teams

1. Encapsulate existing Content, Apps, SharePoint Pages ..etc. In Teams.
2. Build your Apps within Teams.
3. Use Visual Studio (or VS Code) and Teams Toolkit SDK

Existing PowerApps, or Solutions (SharePoint Modern Pages, or other Web Applications)

- If you have already invested in a Power Platform solution, and your aim is to expose your application via Teams and take advantage of the high usage of Teams Clients, you can “Encapsulate” your application in a custom Teams Application.
- The app can be published to your Organization Store
- It can also be pinned to the left rail for all users (enforced, using Teams Policies).
- Any update to your application from Power Platform will available in Teams (it runs as an I frame).



Option 1 – Pros

- This method is very simple and does not require much work, especially when you have existing applications that you want to surface quickly in Teams and take advantage to the increase usage of Teams.
- One of the biggest Pros is that you will be just wrapping your existing applications in Teams, and the complexity level of such application is very minimal.
- You will be building (Your Application Manifest) once only, any new updates to your existing business application will just load in the previously built Teams Application.



Option 1 – Cons

- When it comes to the Cons of this approach, there are no much cons here, except :
- Single Sign on can be tricky to configure especially with non-MS Solutions.
- If you are also planning to include web pages (SharePoint modern pages or business web applications), and these last have a link which opens in a new tab, your users will have to switch back to Teams to continue their journey with your application.
- Such applications can only be deployed to your organisation store.





Demo Time !



Build for Teams in Teams !

- Using PowerApps App for Teams, you can start creating your Application right from Teams.
- This is an ideal scenario if you want to build a new business solution and you know that “most” of the users will access it via Teams
- It involves using PowerApps for Teams and has some particularities

Option 2 – Pros

This option is quite simple and has some great advantages such as:

- Ease to use, build, test and publish
- The publishing process is very simple and requires less fiddling compared to the previous approach.
- It is native to Teams, so the application UI / UX are very close to Teams' Branding.
- When building business applications for Teams using the current Teams layout, your users will require very limited or no training.



Option 2 – Cons

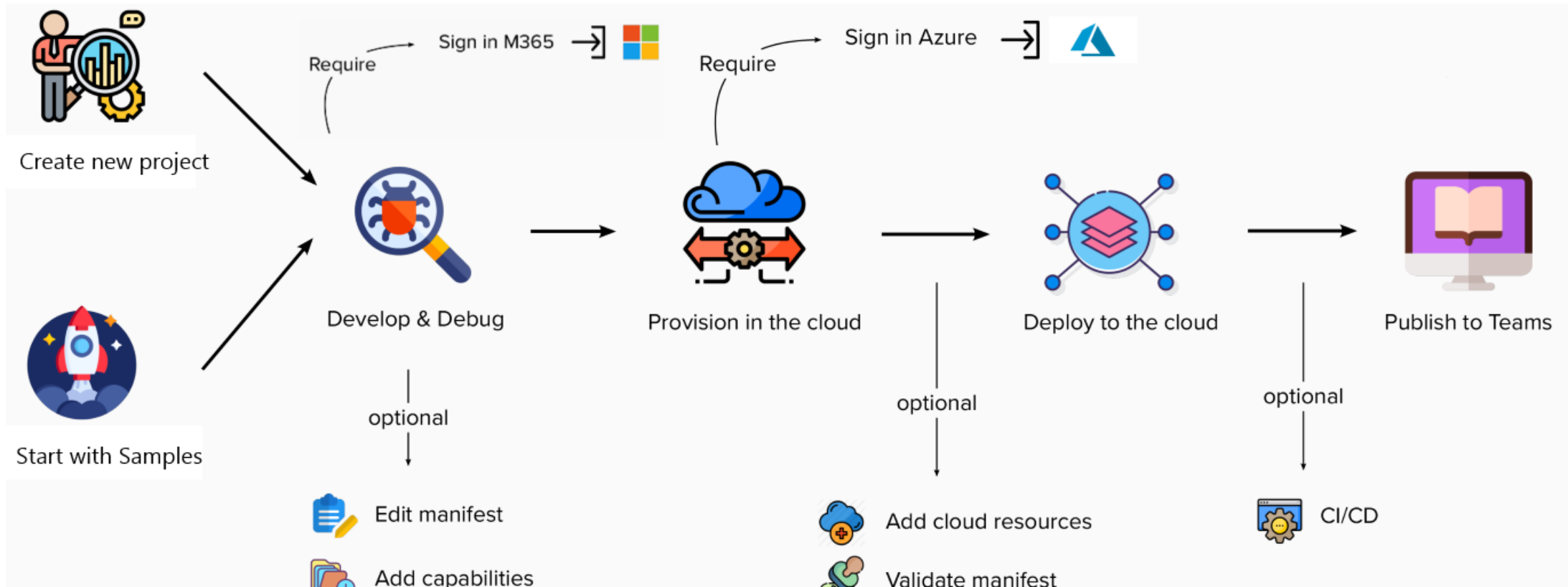
While it is very easy to build, test and deploy, this option has couple of pitfalls:

- Limited to Data Verse only, which means Premium Licences are required for all your users.
- These applications can only be deployed to your Organisation Store at the moment.
- No Co Authoring available yet.





Demo Time !



Custom Teams applications with VS Code.

- If neither option 1 or 2 are suitable to you, and you have complex requirements or you just want to build Teams applications in order to publish to the Microsoft Store or Monetize, you are left with only one option !
Build you custom application with Visual Studio (Or VS Code).

Option 3 – Pros

This last option is meant to be used by pro developers with Node.js or C#.

- One of the main pros of this approach is that there are no limits to the services you can use to build your applications. You can combine M365 and Azure Services, in addition to third party services.
- It also allows you to package the Teams Custom Application and distribute them or monetize them at a larger scale (Teams Store).



Option 3 – Cons

Obviously this approach is more complicated and required advanced knowledges.

The build/test and publish time can be much longer.





Additional
resources

Want to learn more ??

- **Teams Toolkit Overview -->**
<https://learn.microsoft.com/en-us/microsoftteams/platform/toolkit-fundamentals?pivots=visual-studio-code>
- **Prerequisites for creating your Teams app-->**
<https://learn.microsoft.com/en-us/microsoftteams/platform/prerequisites>
- **Beginners' crash course to build apps for Teams using Team Visual Studio Code -->**
<https://devblogs.microsoft.com/microsoft365dev/beginners-crash-course-to-build-apps-for-teams-using-teams-toolkit-for-visual-studio-code>



Any Questions ?

- Slides available from:
- <https://bit.ly/dynamic365-ugnov>

