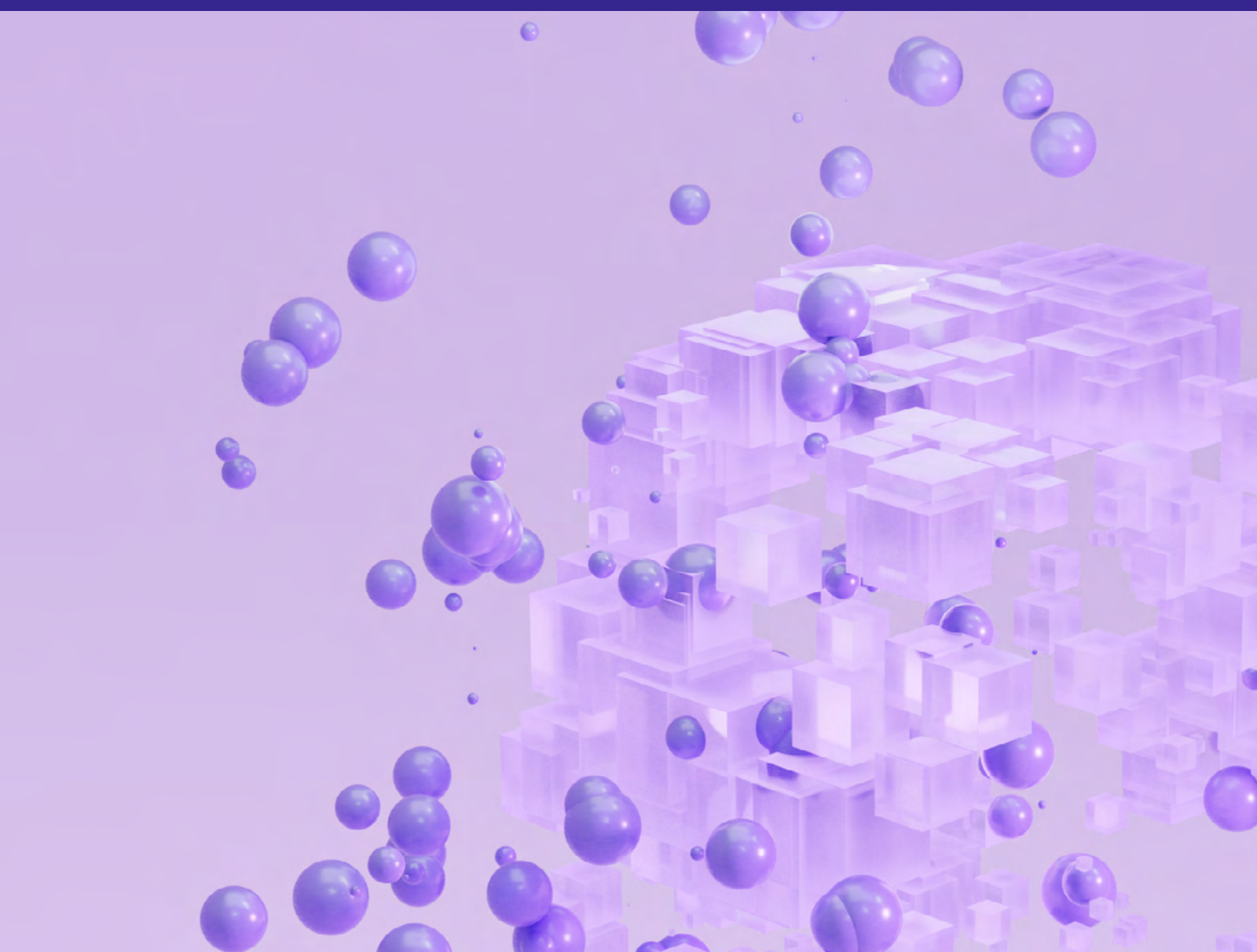
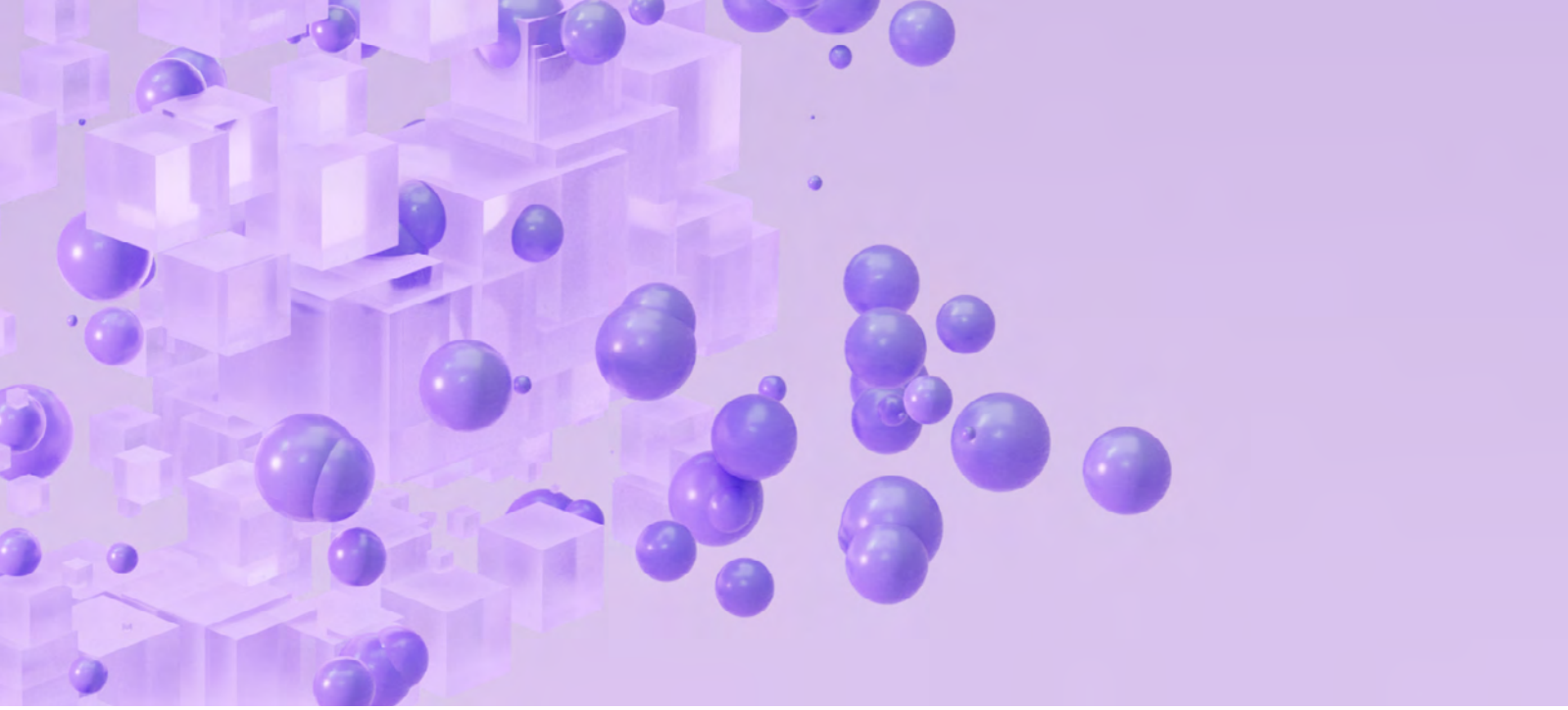
Microsoft 365

Building collaborative apps

with Microsoft Teams and Microsoft Power Platform integration





Contents

Executive summary

In today's hybrid world of work, people have a multitude of collaboration tools and apps that they can use to get work done. There's an opportunity for organizations to create purposeful collaborative platforms that can integrate various apps and provide people with a new way to work without switching between applications and processes.

This integration and accessibility across the organization without having to switch across software platforms gives developers the opportunity to create a new world of collaborative apps. The challenge with IT in the current context, however, is that there aren't enough developers to embrace this change.

Adopting a low-code app development platform allows IT teams to enable their business customers more than ever before. IT application development is moving toward low-code, high-productivity application platforms to minimize cost and support rapidly changing business needs. Instead of depending or waiting on developers, any app maker with business domain expertise can rapidly build collaborative apps using a visual approach, prebuilt components, connectors, and automated processes. Developers can work with app makers, using low-code platform tools to fill the required complex functionality and integration task gaps to build, extend, and scale applications.

Power Platform is a set of low-code tools from Microsoft that enables app makers or developers to build apps quickly with a point-and-click approach to app design. Microsoft Teams allows you to chat, meet, call, and collaborate all in one place. Power Platform makes Teams more powerful by collecting and sharing critical information, automating repetitive tasks, and allowing people to chat with interactive bots. With Teams and Power Platform, you can build apps in weeks, not months. Any app maker can create a proof of concept in a few hours and a workable app in few days. A developer can further extend and scale the app to make it production ready in a week or two. Thus, the Teams and Power Platform integration speeds up the iteration and development of collaborative apps.

This whitepaper discusses the various capabilities of Teams and Power Platform integration for developers. It explains how this integration simplifies and amplifies developers' productivity by helping them:

- Build collaborative apps using the Teams Toolkit for Microsoft Visual Studio and other languages.
- Extend low-code applications using Azure Functions and Azure API Management to create custom connectors to any cloud-based API.
- Publish, manage, and govern various APIs.

- Add Microsoft Azure services like Analysis Services and Cognitive Services to these collaborative apps using custom connectors with Azure API Management to make them more unique.
- Use AI models to automate processes and glean insights from data in Power Platform.
- Build reusable components and work with advanced Power Apps component framework (PCF) features to enhance the user experience while working with data on forms, dashboards, and canvas app screens.
- Get support for application life cycle management (ALM) and automate common build and deployment tasks with out-of-the-box Azure DevOps pipelines and GitHub workflow templates.
- Enhance other developers' capabilities with rich functionality, including Power Platform Build Tools and PowerFx, to enhance developers' speed and effectiveness at work.

Purpose of this whitepaper

This whitepaper is for developers to show how they can rapidly and effectively plan, create, collaborate, and deploy apps built on Teams with Power Platform. It highlights how they can work with app makers in Teams and Power Platform to build collaborative apps using fewer coding tools while scaling and extending these apps with Azure services and other complex business logic.
This whitepaper discusses how developers can experience the full range of development and ALM functionality with rich professional developer tools like Visual Studio, Azure DevOps, and GitHub. They can use their skills with the Power Apps component framework (PCF), Microsoft Power Platform command line interface (CLI), and Visual Studio Code to create value when developing apps.

Building collaborative apps for hybrid work

Hybrid work has become the new normal for companies, and organizations are focusing on this new reality. The way of working has been transformed, and collaboration is at the core of all work and processes.

Collaboration is no longer limited to chat or document sharing to work together; it's more about combining collaboration capabilities with integrated tools and applications that enable people to get their work done faster and more efficiently. This creates an opportunity for businesses to build the next generation of meaningful apps where collaboration is at the core.

Collaborative apps power the new way of work

Hybrid work requires structural changes to how we build and interact with apps. It opens the door to building apps that can seamlessly plug into the collaborative canvas and allow users to work without switching between tasks and tools. We need to build a new class of apps centered around collaboration, enabling synchronous and asynchronous modes of collaboration with real-time meetings, ad-hoc messaging, document collaboration, and business process automation.

This demand has accelerated the need to build a new class of collaborative apps faster and with a higher level of quality and accountability. IT teams are expected to develop apps that can integrate and automate cross-business processes, information, and data. Today, many organizations are already in the midst of digital transformation, and application developers are at the epicenter of this change. This means that organizations are facing ever-increasing backlogs of requested app development and modernization projects. However, development resource constraints have slowed this process, and organizations can't keep up with all these requests. The demand for developers over the next 10 years is growing at a rate that's five times faster than it is for other jobs.[1] Plus, building apps entirely from scratch simply takes too long.

Building collaborative apps and the low-code platform opportunity

All these challenges lead to significant changes in IT leaders' approach to app development and modernization. IT today needs a high-productivity development platform to help them adapt and respond to rapid developments in real time. Rather than turning everyone into developers, employing low-code development platforms can help address these challenges and close the skills gap that exceeds the current enterprise capabilities. Low-code development platforms remove the technical development complexities of automation, integration, and rapid prototyping.

Datapoint

Gartner predicts that by 2024, three-quarters of large enterprises will be using at least four low-code development tools for IT application development and citizen development initiatives. Low-code application building would gather more than 65 percent of all app development functions by 2024.[2]

[1] U.S. Bureau of Labor Statistics: Software Developers, Quality Assurance Analysts, and Testers, June 2021.
[2] The Rise of Low-Code App Development, Forbes, April 2020.

The low-code app development platform: Using the right tools for the job

With a low-code development platform model, any business user can also be a developer and get involved in app development. Instead of depending or waiting on developers, any business user with business domain expertise can build collaborative apps to solve business problems, as traditional development experience isn't strictly required. This business user can become an app maker by rapidly creating a workable app with a low-code development platform. The app maker can envision, design, build, and implement an app—for example, to simplify, automate, or transform tasks and processes.

Low-code application platform tools take a visual approach by automating every step of the application development life cycle. This enables app makers and developers to build the application rapidly and effectively. With the low-code development platform, you can quickly develop new processes or apps with templates, automate workflows based on events, and create data flow controls with variables to manage data across platforms. Prebuilt integrations and API accessibility make it possible to automate business processes. App makers engage with developers for more complex functionality and integration tasks that need strong development skills to build and extend the low-code application. This app development synergy between traditional developers and app makers—so-called "fusion" teams—can help you meet diverse skill sets and cross-functional, cross-disciplinary teams. This facilitates business and IT interaction by getting a working app in front of end users early. App makers can quickly build an app based on business needs and work with developers to fill in the gaps. Users can provide suggestions and feedback on missing functionality or any changes that may be required.

Integrating Teams and Power Platform: Merging a low-code tool with a collaboration platform

Using Power Platform as a set of low-code app development tools and Teams as a collaborative platform, developers and app makers can quickly and easily create collaborative apps that help people avoid having to constantly switch between various apps. Instead, they can spend their time within a team that effortlessly brings together all the relevant information, data, processes, and people.

Power Platform: A set of low-code development tools for all developers

Power Platform is a set of low-code tools that enable any developer or app maker to create collaborative apps with Power Apps, automate workflows with Power Automate, and create interactive reports and bots with Power BI and Power Virtual Agents, respectively—which have all been traditionally done by developers. Developers can work seamlessly with app makers to help solve complex technical challenges using the rich extensibility model provided by Power Platform. Using the native integration with Azure, developers can use their cloud development skills to extend to Power Platform. Power Platform provides a true application development and automation experience. This includes application development and management processes for traditional developers for testing and continuous integration and development, version control, issue tracking, one-click deployment, and much more.

Datapoint

Power Platform lowers development costs by up to 48 percent over traditional coding by increasing the speed of development and reducing the number of resources needed to maintain your apps.[3]

To dive deeper into how to build better apps, you can check out the Transform your business applications with fusion development learning path. It shows you how a fusion development team functions and how a professional developer can enable their app makers to build Power Platform apps and solutions. It also explains how to use pro-code tools and frameworks that are already familiar and comfortable, like JavaScript, ASP.NET Core web APIs, and Azure API Management.

Power Platform drives developers' productivity and agility

With the Power Platform set of low-code development tools, developers don't need to spend time building components from scratch. They can get the required capabilities out of the box to build core application modules, speeding up the process of getting an app to production. Power Platform provides the out-of-the-box modules needed for application development, ranging from data management to sales automation to customer service integration. Power Platform is extensible, providing simplified integrations with Azure services and legacy applications. It enables developers to turn APIs and web services into reusable building blocks. It also offers Microsoft security and governance modules and procedures to ensure that apps built using this platform are safe, controlled, and always protected. ALM integration with Power Platform tools streamlines and simplifies the development process stages for developers—from testing to debugging to deployment. It also accelerates the time it takes to integrate and deploy new tools and technologies into app building.

[3] The Total Economic Impact of Microsoft Power Platform commissioned study conducted by Forrester Consulting, February, 2021.

Microsoft Teams: The perfect platform for building collaborative apps

With 250 million daily active users (and growing), Teams helps people start their day and stay in the flow of work. It's the perfect platform for collaboration. The next step is to bring the apps that users need the most and put them right within Teams, where workers are spending increasingly more time. Microsoft offers a full stack of technologies to build collaborative apps in Teams, making it easy for developers to use their current skills to integrate existing apps into the platform. They can use standard web technologies (along with a few lines of code) and Teams web APIs to integrate these apps into Teams messages, channels, and meetings. Hundreds of independent software vendors (ISVs) like Service Now, Workday, Adobe, SAP, and more are already doing this on today's Teams platform.

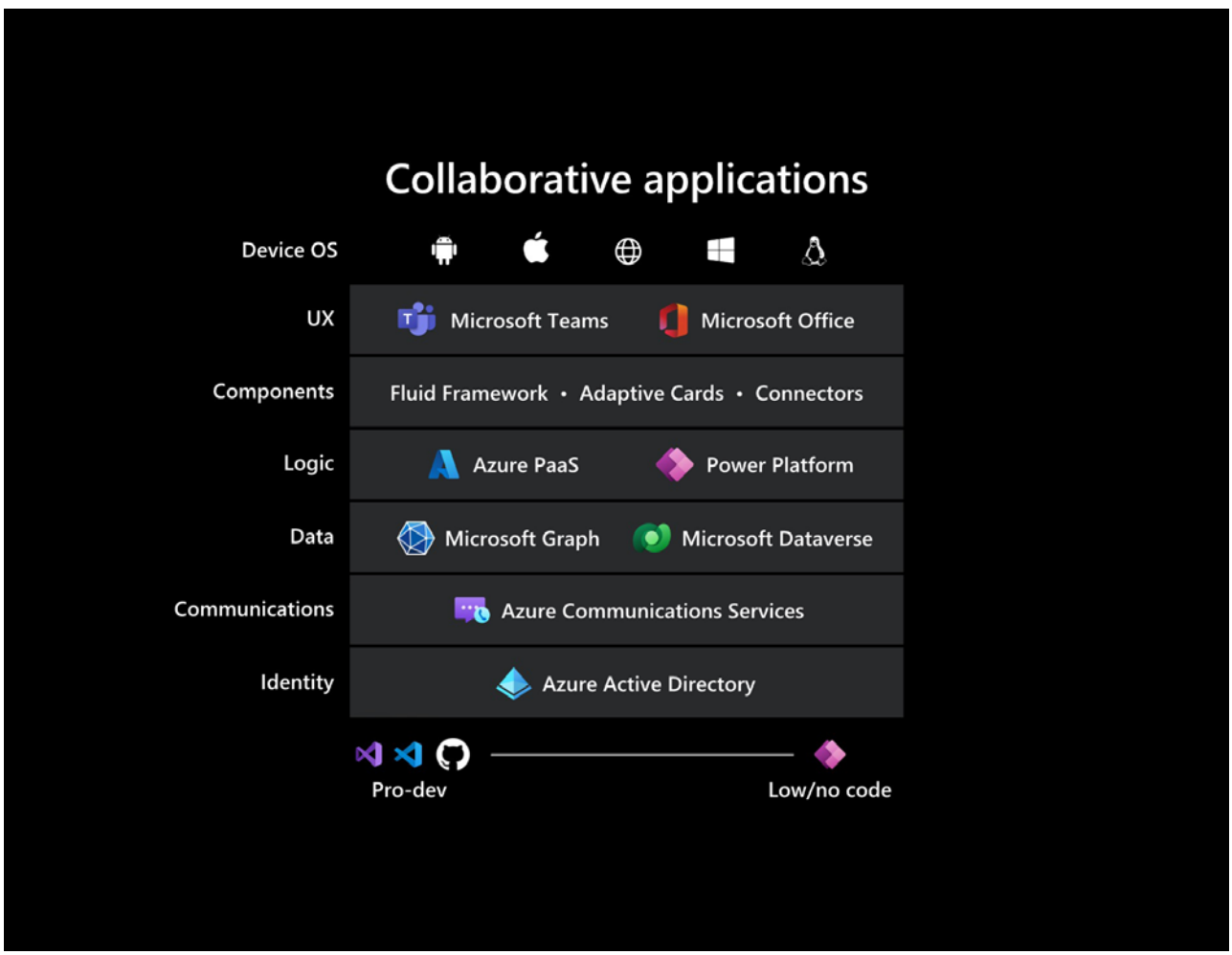


Figure 1: Building collaborative apps in Teams

When you build an app for Teams, it works across Windows, macOS, iOS, Android, Linux, and the web. Teams apps can be built with C# or C++. At the highest tier, we have Teams and Microsoft Office, with collaboration at the center. Teams and Office serve as the interface where users can stay in their workflow with collaborative apps. We have the Fluid Framework and Adaptive Cards on top of the app logic, and we have connectors as a data layer piece. The Fluid Framework is designed to help every developer build better collaborative apps using familiar programming patterns. Connectors keep team apps synced by delivering content and updates from services that are frequently used directly in Teams channels. With connectors, your Teams users can receive updates from popular services like Twitter, Trello, GitHub, and Azure DevOps Services right in their Teams chats. Finally, Adaptive Cards are actionable snippets of content that you can add to a conversation through a chatbot or messaging extension.

App makers can use Power Platform to quickly build a line of business apps and bots, as well as a range of simple to complex workflows with Power Automate. Developers can tap into the full power of Azure platform as a service (PaaS) with things like Azure Functions, Cognitive Services, and Azure API Management to build APIs that are aligned with their business needs and support centralized and scalable management for those APIs. Teams supports Microsoft Graph, where people, their relationships, meetings, documents, what's shared with whom, and what's trending in the organization are captured at the data tier. Then there's Microsoft Dataverse, part of Power Platform with Teams integration. It provides a database in which to store data definitions, tabular data, files, customizations, deployed bundles of customizations, and much more. Azure Communication Services helps developers build custom apps that interact with Teams and participants over voice, video, chat, and screen sharing. Developers use Azure Active Directory (Azure AD) for identity management and access controls and can build their apps on top of it.

Building apps for Teams meetings

Meetings are not static experiences; there are several user interactions and activities leading up to, during, and following meetings. The new meetings extensibility model provides an opportunity for developers to build and customize apps, including the app panel, through in-meeting app signals for notifications. It also allows them to access critical APIs. Developers can enable these capabilities by configuring their Teams app to be available in meeting scope within their app manifest. They can update their app manifest and use the context properties to determine where their app must appear.

Microsoft also continues to add new features to Teams for developers to build richer meeting experiences into their apps. One such feature is shared stage integration, which provides developers with access to the main stage in a Teams meeting through a simple configuration in their app manifest. New meeting event APIs automate meeting-related workflows through events such as meeting start and meeting end. In addition, the Together mode extensibility lets developers create and share their custom scenes for Teams meetings. Media APIs with resource-specific consent helps developers and app makers build scenarios like transcription, translation, note-taking, insight gathering, and more.

Teams and Power Platform integration for building collaborative apps

Teams enables people to collaborate at the forefront of app design. While in the Teams interface, app makers can create and share Microsoft Power BI reports and Power Apps, create automated workflows between apps and services, and create bots and digital assistants—all without ever leaving Teams. Developers can build custom connectors with Azure API Management and Azure Functions to any Microsoft-hosted third-party, legacy, or line-of-business apps. This connector can be published via the web UI of Power Apps or Power Automate, the Power Platform CLI, or an API call from custom code. Apps built in or for Teams can use custom connectors. Apps built with Power Apps that use these new connectors can be deployed to Teams at no additional cost. Apps can automatically be added to Teams. Other members can easily install them as they would with any other Teams apps. IT teams can manage access to individual apps and use audit logs to investigate installation activity, all from the comfort and familiarity of the Teams interface.

With an embedded app studio in Teams and Dataverse for Teams, anyone can now build, customize, deploy, share, and use apps—all within Teams. Dataverse for Teams—the built-in, low-code data platform—backs Teams capabilities to create apps, bots, and workflows. It provides relational data storage, rich data types, enterprise-grade governance, and one-click solution deployment. With Dataverse for Teams, Teams users can find and install custom, ready-to-use solutions from the Teams app store that showcase common scenarios across industries. The Teams Toolkit for Visual Studio lets developers use their existing web development skills to build cross-platform Teams apps against any back end.

Teams and Power Platform app development vs. traditional app development

In traditional development, the process starts with requirement gathering, then goes to the design phase, then to development, and then to release. This development cycle takes a long time, and it takes a while until the user can see the working or prototype app. In addition, there could be gaps between what the user initially envisioned and what the app developer created. In traditional development, the developer needs to manually write code for every component of integration, security checks, and more. Developers have to spend considerable amounts of time integrating services, carrying out integration tests, and applying solid security around the integration. In fact, they need to design and create their own security models for applications. Thus, a significant amount of time is spent developing minimum viable products for users.

The Teams and Power Platform integrated development platform provides the required building blocks to develop collaborative apps. App makers and developers can skip the steps of repetitive code and access the required connections, connectors, and standard templates through a wide range of out-of-the-box components. Power Platform removes the complexities of integration via custom connectors and handles most of the security for developers as well. It provides access to more than 400 data sources out of the box for Teams apps to systems like SAP, ServiceNow, Workday, and Salesforce to connect Power Apps. However, suppose developers require access to a data source where no built-in connectivity is available. In this case, they can easily and quickly develop a custom connector.

This integrated development platform brings high velocity with a full range of out-of-box components needed to efficiently roll out a new app. Developers need to spend time on the code that matters. This means that low-code and high-productivity data modeling, data integration, security, ALM, enterprise-grade governance, and much more are available out of the box for developers to use in their app development. For example, granular and fine-tunable data loss prevention (DLP) controls are built into Power Platform to block specific connector actions or connection endpoints with just a few clicks. This helps developers strike a balance between developer productivity and app protection.

All these capabilities in the Teams and Power Platform development environment speed up the iteration and development of collaborative apps. With Power Apps, app makers or developers can quickly create a prototype of the app because Power Platform enables a what-you-see-is-what-you-get (WYSIWYG) development experience. They can see and visualize the actual working app very early in the development process. If new requirements arise, new features can be added to the next version. With Power Platform, you can have a proof of concept up in a few hours, create a workable app in few days, and have a production-ready app within a few weeks. Developers don't have to worry about the nitty-gritty work of security, governance, and integration. Power Platform takes care of all of this.

Developing collaborative apps with Teams and Power Platform

When these two technologies—Teams and Power Platform—merge, they provide a powerful tool for developing collaborative apps. With the full Teams integration, Power Platform takes collaborative app development to the next level. Let's see how this integrated platform enables any developer to easily develop collaborative apps.

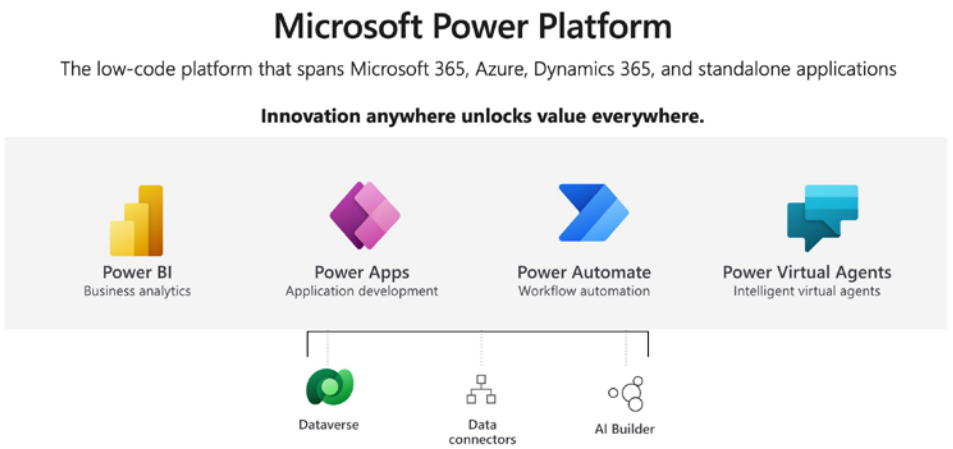
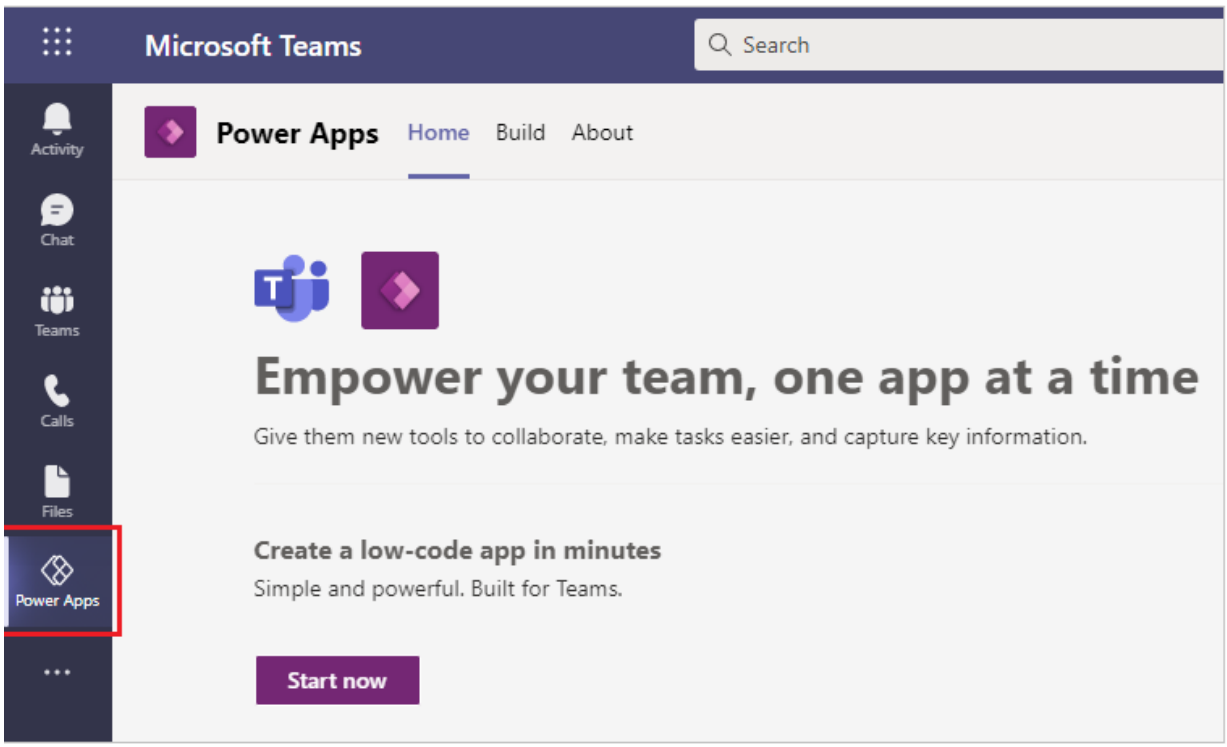


Figure 2: Microsoft Power Platform

Create apps in Teams with Power Apps

With Power Apps in Teams, app makers can quickly create and edit apps within Teams—and publish and share them for anyone on the team to use. If they create an app using Power Apps in Teams for the first time, a new Dataverse for Teams environment would be created.

To create an app in Power Apps, app makers need to select Power Apps and choose which team to host. Then, they need to create, name, and save it. They can also share Power Apps to users beyond the team. When they choose to create an app, they're taken to the Power Apps Studio canvas app builder. They can use Power Apps Studio to design, build, and manage their canvas app. Power Apps Studio contains a data platform that provides an easy way to create tables to store and add data to new tables as needed. There's also an app checker that looks for potential issues as you're building your apps.



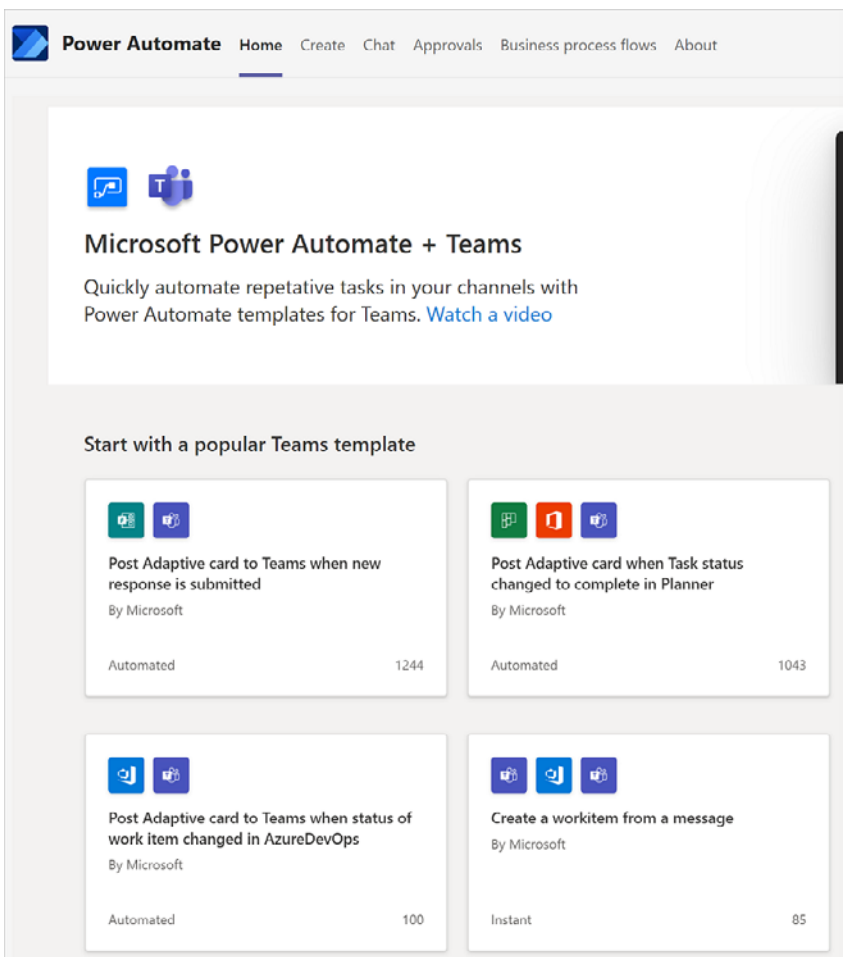
Developers can build more complex Power Apps with multiple data sources, business logic, and advanced controls. And if they need more custom logic on the back end, they also have all the coding environments available in Azure, including APIs and managed services like Azure Functions, Kubernetes, Cognitive Services, and many more. They can build APIs as serverless functions, integrate Power Apps as part of continuous integration/continuous delivery (CI/CD) pipelines, and publish Power Apps to Teams for increased discoverability. Check out the new learning path to transform your business applications with fusion development.

Create flows using Power Automate in Teams

The Power Automate app lets app makers manage and create their workflows directly within Teams.

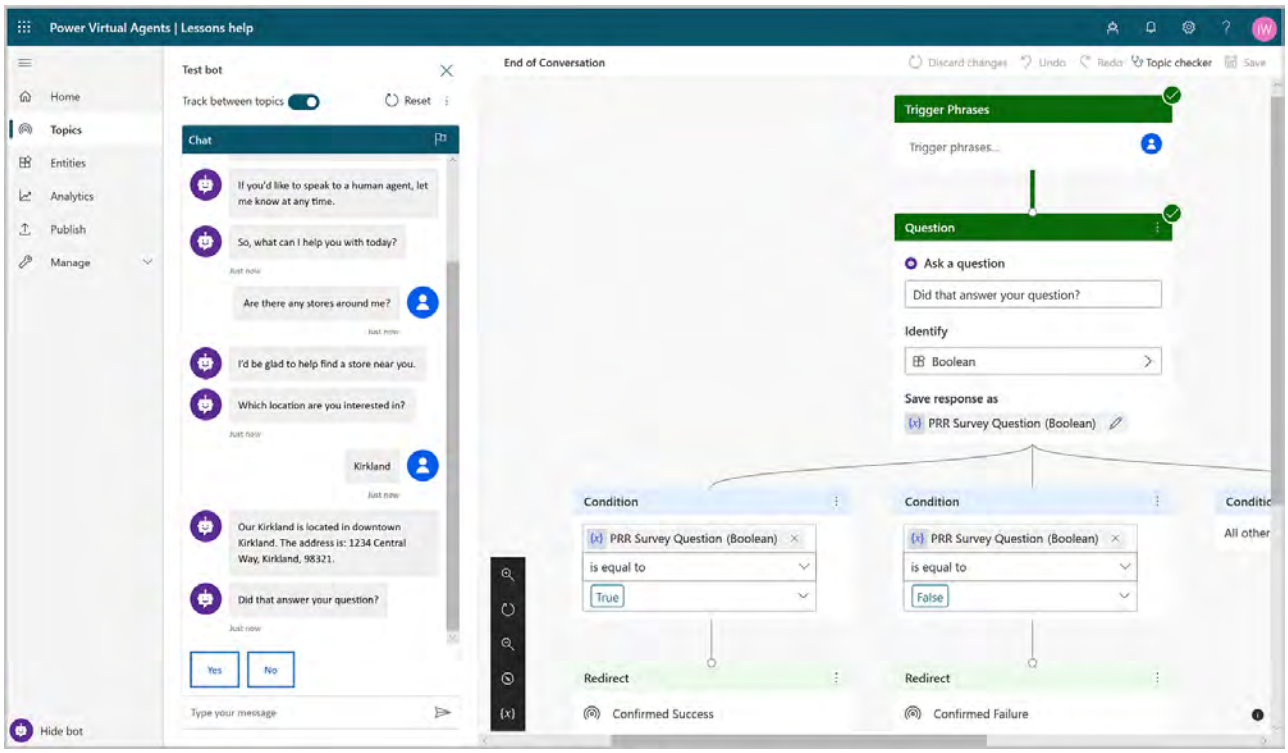
All they need to do is add Power Automate to Teams and create a new flow via templates. If they want complete control over the flow, they can create a blank flow and start from scratch. When they select **Create from blank**, it brings up the full Power Automate designer experience within Teams, where they can create a fully customized flow. The flows can be automated, instant, or scheduled. The difference between these flows is how they're run. For example, you can manually trigger the flow in Instant flow and choose the actions from dozens of connectors. You can use the selected message in the Teams connector to trigger an instant flow directly from within Teams.

App makers can add automatic alerts, notifications, and more—all without leaving Teams. They can also aggregate and automate their team's approval processes in the Approvals tab. Plus, with the Flow bot, they can quickly trigger scheduled flows. Developers can build custom connectors and use the built-in HTTPS connector to make web calls to other APIs as well.



Design bots with Power Virtual Agents

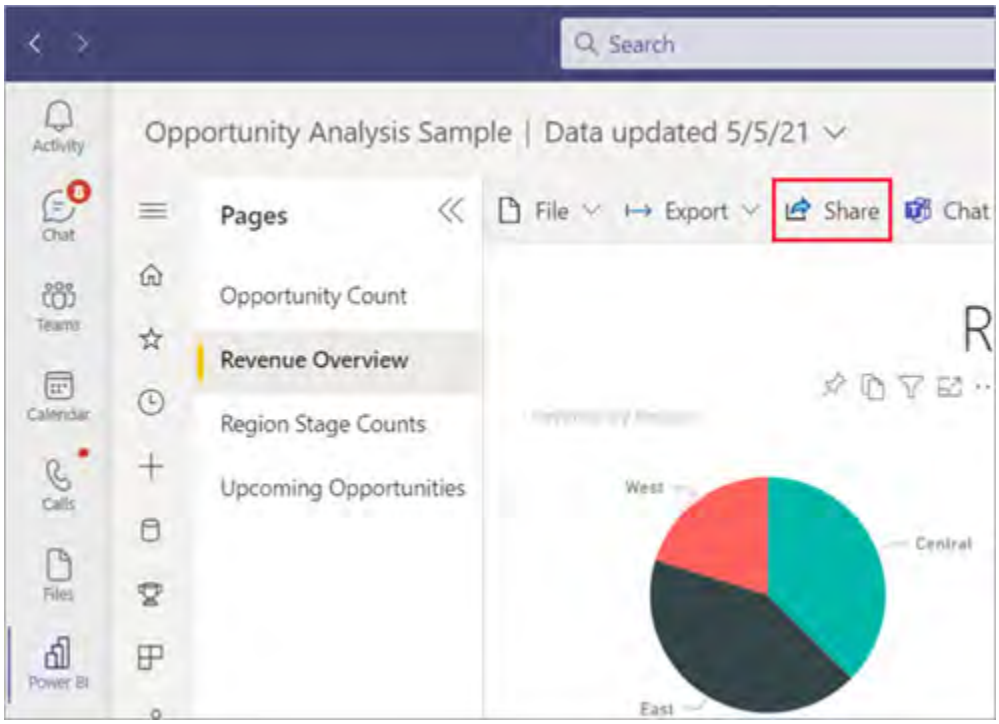
The Power Virtual Agents (PVA) app is available within Teams. It helps any app maker easily create chatbots to engage with the team, address frequently asked questions, and save time by triaging common IT questions. The PVA platform delivers a whole AI-based environment that allows chatbots to work. The only thing that's required to deliver is the knowledge and logic for bot behavior. Any app makers can add the Power Virtual Agents app from the Teams app store and start creating, authoring, testing, and publishing their bots directly within the Teams interface. They can use templates created by the PVA team to help create and customize bots to their needs.



There's an integration between PVA and other Power Platform products in Teams. For example, you can add actions to a bot by calling a Power Automate flow. Flows can help you automate activities or call backend systems. The data can be passed into the flow to handle actions such as lookup data, write data, return data from the database, and return data to the PVA.

Create powerful data visualizations with Power BI

Power BI is an assortment of cloud-based services that help your organization collect, manage, and analyze data from various sources. Any app maker can add the Power BI app to Teams. Power BI brings the entire basic Power BI service experience to Teams. It enables app makers to quickly discover, search, and discuss data without ever leaving Teams. App makers can create, view, edit, and share dashboards, reports, and apps. They don't need to hop into the browser—they can just stay in Teams, use chat, add reports to messages, and do lots more. And there's a feature in Power BI within Teams that you don't see when you view the Power BI service (app.powerbi.com) in a web browser.



Dataverse for Teams

Dataverse for Teams is a built-in, low-code data platform that empowers app makers to build custom apps, bots, and workflows using Power Apps, Power Virtual Agents, and Microsoft Power Automate, respectively—without ever leaving the Teams interface. It's automatically created for the selected team when you create an app or bot in Teams for the first time or install an app from the app catalog for the first time. It can store, manage, and share team-specific data, apps, and flows. Because data and metadata are both stored in the cloud, apps are simple to manage and administer. Additionally, a built-in security model allows you to control tables and functionality based on your organization's users' roles.

Bring AI to your apps with Microsoft AI Builder

AI Builder is a Power Platform capability for Teams that brings the power of AI through a point-and-click experience. It's directly integrated into Power Apps and Power Automate. App makers can use AI models created using AI Builder in canvas and model-driven apps through the formula bar or builder component to add intelligence to your apps.

Add data connectors to bridge data and actions

The built-in connectors from Power Platform enable the user interface to connect to many data sources, such as SQL Server, Microsoft SharePoint, Microsoft Excel, Microsoft Dynamics 365, Twitter, Oracle, and hundreds of others, without having to write a ton of code. There are hundreds of connectors already available, including SharePoint, SQL Server, Office 365, Salesforce, Google Services, and Twitter, enabling app makers to connect data and actions without code. Developers can also create custom data connectors that app makers can consume in their apps and workflows.

Developer productivity with Teams and Power Platform

The previously discussed Power Platform features and capabilities are great to build collaborative apps, but sometimes you need additional and complex functionality that only developers can supply. As a developer, you need to understand the gaps between what can be accomplished through visual configuration versus code writing in Power Platform. Sometimes, existing features might not provide the functionality needed to meet a requirement. Power Platform provides rich tooling where developers can extend the standard functionality by using code. Developers can work seamlessly with app makers to help solve complex technical challenges using the rich development and extensibility model provided by Teams and Power Platform components. This is where the fusion development approach with Power Platform combines the worlds of app makers and developers in building, scaling, and extending collaborative apps to meet business objectives.

Power Platform provides true app development and automation experience for developers. This includes app development and management processes for developers for API management, testing and continuous integration and development, version control, issue tracking, one-click deployment, security, governance, and much more. They can solve complex challenges using a rich extensibility model provided by the components of Power Platform.

Professional developer app development extensibility

Developers extend low-code apps with flexibility powered by Azure services. Using the native integration with Azure, developers can use their cloud development skills to extend Power Platform with Azure PaaS and data services. The Microsoft stack with Power Platform offers a rich extensibility model that taps into other ecosystems and capabilities. We want to make sure that we're offering the right tools for the right tasks to the right people. The following is a high-level overview of each of the core components of the Microsoft stack and their key extensibility points.

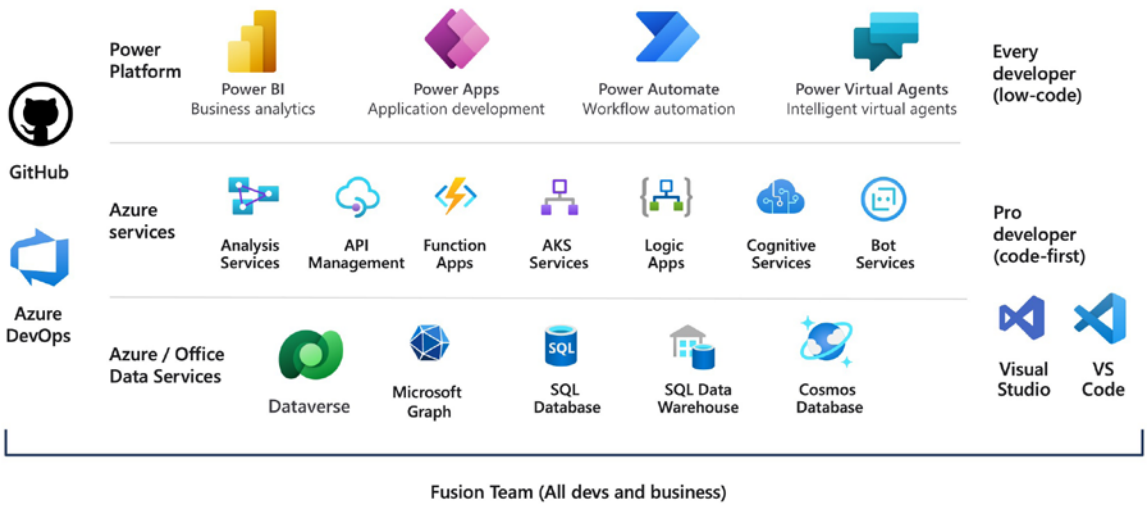


Figure 3: The core components of the Microsoft stack and their key extensibility points

At the top there's Power Platform, which offers low-code development. It has drag-and-drop functionality with an easy-to-use UI to help develop business apps. Both developers and app makers can build feature-rich, custom business apps while writing very little code. This reduces the app load on developers, promoting a new fusion collaboration and co-development model.

In the middle, there's the Azure Services as a workload engine. This is where developers can build complex business logic and custom functionalities to elevate them into Power Platform. They can focus on more complex capabilities through high-value work such as building APIs with Azure Functions, Microsoft Analysis Services, Kubernetes, Azure Logic Apps, Azure Bot Service, and Azure Cognitive Services. Developers can use Azure API Management and Azure Functions to scale and extend the apps.

Next, there's the Azure and Office Data service, which can store data. Dataverse is the underlying data platform for Power Platform that contains the core functionalities, such as server-side logic (plug-ins and workflows), business process flows, a highly sophisticated security model, and an extensible platform for developers to build apps. Power Platform also supports SQL Server, Cosmos DB, Microsoft Graph, and Azure Synapse Analytics as a managed Azure cloud database to simplify data management and optimize utilization.

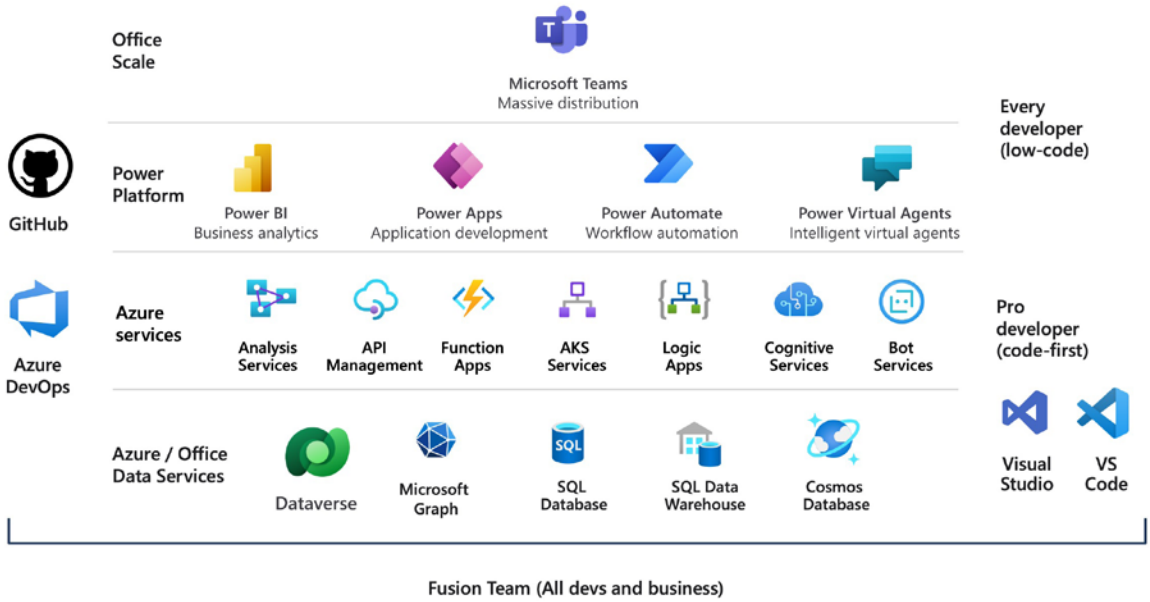


Figure 4: Developers and Microsoft Power Platform equal no limits

Power Platform makes it easy for developers to build and extend Power Platform apps and solutions with native Visual Studio and Visual Studio Code integrations. Developers can also simplify the ALM process using Azure DevOps. Power Platform Build Tools are a collection of Power Platform-specific Azure DevOps tasks that eliminate the need to manually download custom tooling and scripts to manage the application life cycle of apps. With GitHub Actions for Power Platform, developers can create workflows in their repository to build, test, package, release, and deploy apps; perform automation; and manage bots and other components built on Power Platform.

Power Platform, Teams, and Azure API Management (APIM)

Developers can use Azure Functions and Azure APIM to create custom connectors to connect to any cloud-based API, including Microsoft cloud-hosted data sources. Azure APIM allows developers to publish, manage, and govern various APIs. Developers can use Azure APIM in Dataverse for Teams. With Azure APIM, they can publish their backend services as APIs. Any app maker can easily export these APIs to Power Platform (Power Apps and Power Automate) as custom connectors for Dataverse for the Teams environment for consumption. App makers can build apps that use these custom connectors via Power Apps or Power Automate hosted in Teams with no additional licensing costs.

Azure APIM and Power Platform

1. Developer: Build a custom connector through Azure Function of custom API Management.

2. Developer: Publish the custom connector to the Power Platform.

3. App maker: Use the custom connector in your Power Apps solution.

4. App maker: Distribute your Power Apps solution in Teams.

To do this, you need to have an Azure APIM instance. Developers can build and secure your API in Azure APIM. They can describe the API, define specifications, and publish the connector. Azure APIM can export the APIs to Power Apps and Power Automate through the wizard experience. This API is then available as a custom connector in a Power Apps or Power Automate environment.

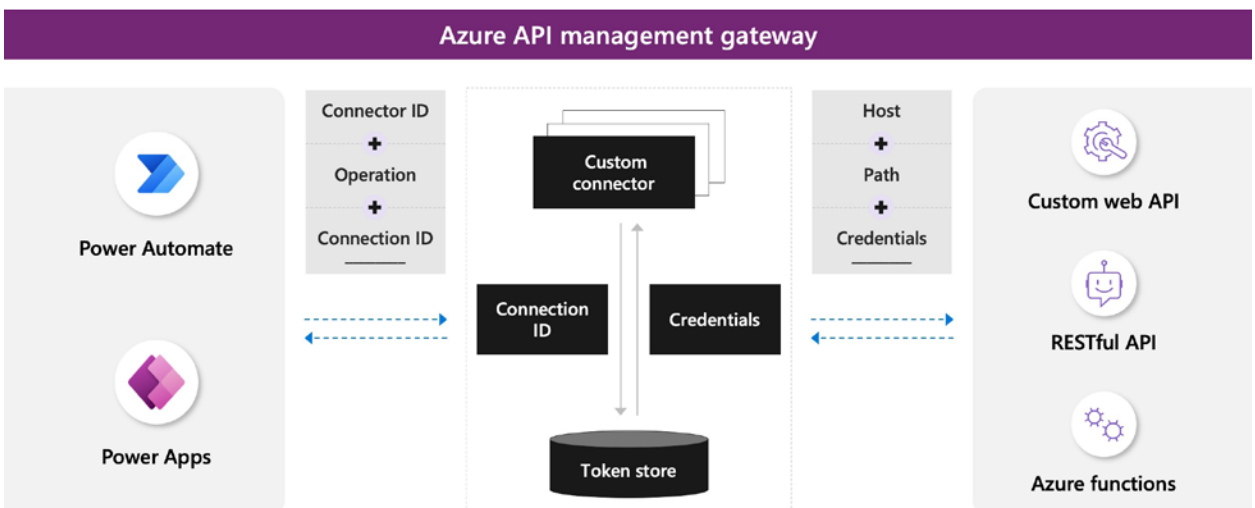


Figure 5: Custom connectors

Developers can also use the HTTP connector to fetch resources from various web services authenticated by Azure AD, or from an on-premises web service.

Developers can build unique, rich functionalities that differentiate apps using Azure services such as Azure Analysis Services, Azure Functions, Azure Cognitive Service, Azure IoT Edge, and Kubernetes Services and expose them as custom connectors. This further amplifies their pro-code components and unlocks access to any Microsoft cloud-hosted services with just a few clicks to empower app makers to build components. For more information on how to export APIs from Azure APIM, visit our page on how to export APIs from Azure API Management to the Power Platform.

Support for the whole development life cycle

Power Platform has a rich spectrum of tooling capabilities that provide developers with excellent app development and automation experience. Developers can use their existing skill sets and code investments. For example, the PCF allows developers to use common frameworks like React, which uses a library of components they already know. They can take these React controls, put them into solutions, and use them as part of Power Platform and Power Apps.

Power Platform CLI support

For fusion team collaboration, developers can transpile apps into human-readable source code—a friendly format that enables apps with enterprise CI/CD pipelines to deploy. The Power Platform CLI empowers developers and ISVs to perform various operations in Power Platform related to environment life cycle features, authenticate, work with Dataverse environments, solution packages, portals, code components, and more. Developers can test and validate using command-line tools with live debugging capabilities.

Power Platform tools for Visual Studio and Visual Studio Code

Developers can quickly build code and custom connectors using Visual Studio and Visual Studio Code extensions to extend Power Platform. Power Platform Tools for Visual Studio support the rapid creation, debugging, and deployment of plug-ins. They also include developing custom workflow activities, web resources, and integration technologies like Azure Services endpoints and webhooks.

The Power Platform extension brings the Power Platform CLI to work within the context of the Visual Studio Code editing environment. It allows developers to use the Power Platform CLI from within their integrated development environment (IDE) and to interact with app makers without needing to exit their code-first integrated development environment. The Visual Studio Code extension provides the ability to create solution artifacts, package artifacts, and work with Power Apps portals and PCF.

Integrated multi-experience development capabilities with Azure Services

Azure provides an extensibility surface for hosting many different types of logic and functionality. Power Platform is an integrated development platform with Azure capabilities. Azure includes a variety of prebuilt services that you can access via Power Platform. Dataverse for Teams extends with Azure storage, Azure Cosmos DB, Webhooks, event hubs, and much more.

Developers can create solutions on Power Apps with Azure Services using prebuilt user experience (UX) components and simple frameworks. They can easily add Power Apps AI capabilities with no machine learning expertise or add Azure elements, such as Cognitive Services. Bot designers can use Azure development tools like Bot Framework Composer to create custom dialogs and directly

add them to Power Virtual Agents bots. This provides a simpler way to extend bot capabilities with custom code. It doesn't require additional Azure hosting, deployment, or billing complexities. They can also add AI using the built-in AI builder or expand beyond that into Azure Cognitive Services and RPA. Developers can expand the physical and digital world with mixed-reality capabilities in Power Apps. All these services come with consistent, well-documented APIs, so they're easy to incorporate into apps.

ALM with Power Platform

Power Platform helps streamline ALM using built-in support for CI/CD. Developers can implement ALM using Power Platform with out-of-the-box Azure DevOps pipelines and GitHub workflow templates. It enables them to rapidly package and deploy solutions across environments and tenants. They can also define unique governance models for app makers' apps, mission-critical apps, and more. Both professional developers and app makers can implement ALM using Power Platform with out-of-the-box Azure DevOps pipelines and GitHub workflow templates. This enables them to rapidly package and deploy solutions across environments and tenants. They can define unique governance models for app makers' apps, mission-critical apps, and more.

Power Platform Build Tools automate standard build and deployment tasks related to apps built on Power Platform. This includes synchronizing solution metadata and various Power Platform components, including canvas apps, model-driven apps, UI flows, and virtual agents. Build artifacts can be generated and deployed to downstream environments. In addition, static analysis checks can be performed against solutions using the Power Apps checker service. Power Platform Build Tools tasks can be used with any other available Azure DevOps tasks to compose your build and release pipelines. Pipelines that teams commonly put in place include Initiate, Export from Dev, Build, and Release.

There's also GitHub Actions for Microsoft Power Platform, which have the same capabilities as Azure DevOps. With GitHub Actions for Power Platform, developers can create workflows in their repository to build, test, package, release, and deploy apps. Developers can create software development life cycle (SDLC) workflows or use preconfigured templates with GitHub Actions to develop, test, and deliver Power Platform solutions. They can use GitHub Actions for Power Platform and any other available GitHub Actions to compose your build and release workflows.

Build custom components with the PCF

The Power Apps PCF provides developers with the flexibility to create any UX component to support a fully customized app experience. Components are reusable and solution aware, thus easily sharable, and can plug into any apps. Developers can reuse their current intellectual property and skills, as this framework is based on standard web technologies (such as TypeScript/JS, CSS, and HTML5). These custom controls can be built from scratch in JavaScript and TypeScript, or they can use common development frameworks like React and Angular. App makers can use the custom control in model-driven or canvas apps. These code components can enhance the user experience for users working with data on forms, views, dashboards, and canvas app screens. They provide access to a rich set of framework APIs that expose capabilities like component life cycle management, contextual data, and metadata.

Microsoft Power Fx as the low-code programming language

Power Fx is the new low-code language for expressing logic across Power Platform. It's a general-purpose, strong-typed, declarative, and functional programming language inspired by Excel. App makers and developers can create apps that typically connect to various data stores and use Excel-like expressions for business logic. Power Fx offers imperative logic when needed. Spreadsheets don't typically have buttons that can submit changes to a database, but apps often do. The same expression language is used for both declarative and imperative logic. Power Fx binds objects together with declarative spreadsheet-like formulas. For example, imagine the "visible" property of a UI control as a cell in an Excel worksheet with an associated formula that calculates its value based on the properties of other controls. The formula logic recalculates the value automatically, like a spreadsheet does, which affects the visibility of the control. Power Fx will be made available as open-source software. It's currently integrated into canvas apps, where you can experience it today.

Rich analysis and debugging tools for developers

The Power Apps solution checker promotes high-quality, model-driven apps by helping app makers follow best practices when they customize and extend the Power Apps platform. The Power Apps checker performs a static analysis of your Power Apps solutions to identify performance and stability risks and provides detailed recommendations to remove these risks from your solutions. You get access to rich, detailed reports listing identified issues, severity, locations, and sometimes line code, with links to detailed prescriptive guidance on fixing the problem.

The Power Apps checker web API provides a mechanism to run static analysis checks against customizations and extensions to the Dataverse platform. It's available for app makers and developers to perform rich static analysis checks on their solutions against a set of best practice rules to quickly identify problematic patterns.

Security and governance controls for developers with Dataverse

Dataverse, the underlying data platform for Power Platform, handles security from user authentication to authorization, allowing users to perform data and service actions. Developers can use the existing functionality of Dataverse to help speed up development. They can focus on building a workable solution rather than divert time to create a sophisticated security model out of the box. It includes a pre-built set of constructs for developers that allow the configuration of an application security model. Dataverse uses Active Directory's identity and access management mechanisms to ensure that only authorized users can access the environment and its relative data and reports. Dataverse natively supports encryption of the received, sent, and stored data because of its Azure nature. Dataverse uses role-based security to put together a collection of privileges that cover both data access with row-level granularity and business functions that can span multiple tables. Dataverse also has a column-level security feature to allow for more granular control of security at the column level. Developers can use fine-grained controls in their applications, such as blocking specific connector actions or connection endpoints.

Dataverse also includes the manager hierarchy and the position hierarchy as a security construct that can be used for hierarchies. With the manager hierarchy, a manager must be within the same business unit as the report or in the parent business unit of the report's business unit to have access to the report's data. The position hierarchy allows data access across business units. These can help build more complex security models. Developers have the flexibility to use one or several of these features when building out their solutions. Developers can also automate security tasks using the API.

Supporting developer plans to build and test Power Apps, Power Automate, and Dataverse

The Power Apps Developer Plan offers a free development environment to develop and test apps, including premium and custom connectors, Dataverse, and additional developer environments. The Developer Plan creates a new Power Platform environment, separate from the environments created within Teams. You can create apps and flows without writing code, with full-featured Power Apps and Power Automate development tools, and easily share and collaborate on these solutions with other team members. With Dataverse, you can use a fully managed, scalable data platform, including common business app actions. You can export the solutions you create in your developer environment and publish them on Microsoft AppSource so that your customers can test-drive them. This developer plan is intended for development and test use only. A paid plan is required to deploy or run solutions in a production environment for production use.

Developers partner with IT for deployment and governance

Developers can connect and partner with IT to deploy and govern Power Platform solutions, which can help them address IT administration and governance concerns in advance.

Dataverse for Teams honors Power Platform's existing data governance paradigms and enables access control in the Teams admin center. The Power Platform admin center provides more detail, including monitoring dedicated capacity utilization and DLP policies. Dataverse for Teams greatly simplifies environment life cycle management and user security role management by aligning to Teams constructs. All identity is managed through Azure AD, enabling rich policies and multifactor authentication. The Teams admin center allows administrators to manage any apps available to the organization through permissions policies. You can allow users to install all apps, no apps, or anything in between on an app-by-app basis. The Teams admin center also provides full audit logs, usage analytics, data loss prevention, security, and data management.

The Power Platform admin center provides a unified portal for administrators to manage environments and settings for Power Apps and Power Automate. They can view Dataverse analytics right from the Microsoft Power Platform admin center to quickly view adoption and user metrics for the organization. The Power Platform admin center brings role-based action controls to a central location, allowing administrators to manage user permissions and set appropriate roles without switching between management interfaces.

Next steps

Get started with Teams and Power Platform today. Join the Power Platform Community to connect with like-minded peers and product experts on Power Platform.