

05242023 Build Rajesh Jha Keynote

**Build 2023**

**Rajesh Jha**

**Archana Saseetharan, Yina Arenas, Wamwitha Love**

**Wednesday, May 24, 2023**

**RAJESH JHA:** Thank you, thank you. Thank you. I'm Rajesh Jha, and I lead Experiences and Devices as Microsoft. I'm just so incredibly excited to be here today to discuss how we can shape the future of work with AI. And then Panos is going to join me later and he's going to talk about the future of Windows.

So yesterday you heard Satya, Kevin and Scott Guthrie talk about the Copilot stack, the AI architecture of the future. Today is about bringing that AI stack to Microsoft products that hundreds of millions of people use every single day. Microsoft is going to democratize the opportunity for developers. That means you and us working together to bring this new capability to users around the globe.

Satya talked about it yesterday, and I've been in this industry a long time. I do think we are looking at the next frontier of computing. When I started in the industry three decades ago, a long time ago, graphical user interface was democratizing computing by making it accessible to people around the globe. Millions and millions of people started to be able to use computing, and AI is even more accessible. It's even more powerful, and working with you, we can transform the way work is done in society, letting people use natural language to do their work.

First, I wanted to take a step back and talk about a survey that we call the Work Trend Index. We talked to 30,000 customers, employees and leaders around the globe in 30 different countries. And so we asked them how they thought work was evolving.

And then we also asked, of course, about their perceptions and expectations of AI. And perhaps it won't surprise you that these 30,000 people around the globe, the one thing on the average that showed up was their spending two days out of every workweek coordinating in meetings, communicating.

And so no wonder two-thirds of them, 64%, they said they're not able to keep up with work. They're struggling to innovate. And the leaders at the same time, three out of five leaders, they said their teams were not innovating enough.

And then we asked him about AI. Not surprisingly, one out of two said they worry about what AI means to their job, what it means to their current set of skills, but fully 70% of the same set of people said they want AI, to be able to delegate work to AI, so they can do meaningful work.

That is our collective opportunity. And why am I optimistic? Why together can we make a difference? Because these Microsoft products, they power businesses, small and large around the globe. These are productivity tools, communication tools that people use to get their jobs done.

Bing, 100 million daily active users Bing and Bing Chat. People are finding information. They are learning. They're researching.

Windows needs no introduction. A billion people around the globe are using it every single month. A billion.

Microsoft Teams, 300 million monthly Teams users doing collaboration, communication, productivity, synchronous, asynchronous for this modern world.

And you've been a part of this journey. It's exciting to see the work that has been done by developers. We have 1,900 applications in the Teams App Store. And perhaps as impressive, 100,000 applications have been created by customers, developers creating line-of-business applications to bring productivity and business processes in the flow of work.

So today we are going to talk about the Microsoft 365 Copilot and our opportunity with the Microsoft 365 Copilot to shape work, the future of natural language. Panos is going to talk about the future of Windows and what that means in this AI world.

First, you know, I want to share with you a little bit about how we thought about designing the copilots. You've heard Satya talk about it. We started with the spirit of Copilot advancing the human agency, putting the user at the center, being grounded in the user's context. This is not autopilot. It's about letting people achieve more by having the AI work on their behalf, being grounded in their context.

That is what you're going to see in the design language of all the experiences we build. And we're going to make the design language available to developers as you extend AI, as you build AI applications, so you'll be able to use the same design language.

So to show you our vision with a copilot, let me run a quick video.

(Video segment.)

Now, the Microsoft 365 Copilot is not just about taking the most powerful foundational models. Yes, it is that. We are running some of the most powerful models out of Azure with the full safety delivered by the Azure cloud. But it's about grounding this AI, these foundational models in the user's context. Who do they work with? What do they work on? Their meetings, their conversations, the documents, and that is the Microsoft Graph.

The Microsoft Graph represents the user. It represents their context. That is what grounds the AI, and then the user experience is brought to applications and workflows that people are already using, whether it be in Office or Teams, and as you will see on Windows, that is Microsoft 365 Copilot. It is a system.

Now, there are two capabilities that you're going to see today with the Microsoft 365 Copilot. Of course, you have the in-application assistance and that's about completing the task at hand.

When you're creating a document, whether it be a report or a presentation or a financial model, the AI powered by natural language is right there with the rest of the computing in the application, right by the user to help them complete their job. So then application assistance is exactly what you saw in the video. That's what you would expect to see work.

I'm also excited about the cross-application intelligence layer. This is like ChatGPT for the enterprise. This in-app, cross-application intelligence reasons over the users, all of their programs and applications, their personal data represented by the Graph and the enterprise content.

It brings all these workflows across the myriad of contexts and applications that a user uses. This is what I mean by user the center. It's less about the apps, put the user at the center, and bring everything around them like ChatGPT for the enterprise.

So today, the rest of this conversation, what I'm going to do is I'm going to talk to you about the opportunity we have for developers, and we'll talk about this in four chapters. We'll talk about how you can extend this cross-application intelligence, and how you can plug in the data layer with the Microsoft Graph. How can you extend the in-application assistance? And then finally, we'll talk about developer success. What does it mean, end to end, based on the feedback you've given us over the years?

So, Chapter one. Let's dive in, into the cross-application intelligence layer in Microsoft 365 Copilot. It works across all your applications, your data, your organization's content. It's available where the users are in Teams. It will be in Windows, it'll be in the M365 app.

And so to show you what it looks like, I'm going to invite Archana. Archana, welcome.

(Applause.)

**ARCHANA SASEETHARAN:** Thank you, Rajesh. Hello, everyone. Very good morning.

Now, I'm going to show you how the Copilot is grounded in your company's data, thereby helping you work more efficiently and effectively. For this demo, I'm playing the role of Anika, who is an account manager.

I have an important monthly sales meeting coming up with a customer AdventureWorks. My company uses both Microsoft 365 and Dynamics 365 to manage the AdventureWorks relationship. First thing first, I'd like to know when is my next monthly sales meeting with AdventureWorks?

So I go ahead and ask that question to Copilot. Copilot is grounded in data from Microsoft Graph so it can access my emails, calendar files, conversations and more to help me with such specific information.

I can also ask Copilot to do more than simply fetch information. For instance, I can ask it to summarize all my latest correspondence with the meeting organizer, which it does so by summarizing my emails.

Now, in this email summary, I see some interesting order information and I'd like to learn more. And because Copilot is also grounded in data from Dataverse, which is where my company's Dynamics 365 data is stored, Copilot can help me with specific order details, along with a link to the audit itself that I can then click through and learn more about.

And finally, I can also ask Copilot to take actions on my behalf, like, for instance, to help me draft an email with everything that I just learned, like the email correspondence summary and the sales order activities, so that I can email that to my colleagues. Copilot helps me with this really useful email draft, which of course I can update with more information, personalize it and make it my own, or simply copy and paste and send it via Outlook.

So that's how by helping me find, summarize and synthesize information from all across my business data sources, Copilot helps me work even faster and makes me even more efficient and effective. Thank you. And back to you, Rajesh.

(Applause.)

**RAJESH JHA:** Thank you, Archana.

So to recap, what you saw was a simple natural language interface that captured the user's intent. It didn't matter if the data was in the graph specifically or whether it was in some line-of-business schema like the Dataverse. And so the copilot was able to reason over all of that and then available where the users already are.

Now, let's talk about extensibility. Yesterday you saw that we are going to make the copilot extensible with plugins. I want to talk to you more about what that means. So when you send a copilot a plugin, the M365 Copilot, the copilot is able to get your information's data. You get to control the user experience and the copilot is able to act on the user's behalf.

Now, we've settled on the same open standard as ChatGPT plugins. So OpenAI and Microsoft, we basically are looking at the open standard. You take an open API, all that you need as a developer and open API on manifest, and then you can create a plugin and that plugin works in ChatGPT, it works in Bing, it works in Microsoft 365 and more.

And I think Scott showed you how he was able to create a plugin that works in ChatGPT, and it was about a plugin that reasoned over a product catalog in natural language for the relevant content. And he showed that working in ChatGPT. That same plugin, you will see now working in M365 and Microsoft Teams, and in that copilot experience in Microsoft Teams.

That's what I meant by the entire interoperability of the work that you do. You do the work once to express an open API endpoint, and then you get the affordability of your extensibility into all of these copilots.

Now, here's the great news. That's not the only way to extend the copilot in Microsoft 365, all your existing investments and message extensions. So today, many developers are extending the Team's messaging so you're able to throw your custom conversation types into the flow of conversations.

Now, we've made that same thing work in Microsoft Outlook, and that is also going to work as an extensibility in the copilot. Furthermore, over the years we've worked with many developers to take Power Platform connectors and connect them to all sorts of backend data sources.

Each of those connections are also available as a new skill for the copilot. This copilot is like the operating system for productivity and business process, and we do that together. You build the extensibility through a plugin, whether it be, you know, the open API standard plugin, or a message extension or using the Power Platform connector.

Now, that is why we are excited that we're going to have thousands of plugins available for Microsoft 365 Copilot. It's not only the new plugins you build in the ChatGPT style of plugin, but also the existing message extensions, the existing Power Platform connected to business applications built by three corporate developers that will all be available as plugins for Copilot over the coming months.

Now, I'm also very excited to talk to you about the tooling work that we've done with our colleagues in Visual Studio. We built a Teams Toolkit to make it incredibly easy to go build and debug and deploy these plugins. And who better to show that to you than Archana? Please take it away.

**ARCHANA SASEETHARAN:** Thank you again, Rajesh.

(Applause.)

It's demo time.

Now, I'm going to show you how you can build a plugin to extend Copilot using the Teams Toolkit extension in Visual Studio Code. I already have Visual Studio Code running GitHub Codespaces, open in this machine here.

Like Rajesh said, all you need to build a plugin for Copilot is a REST API that follows OpenAPI specifications. For this demo we are going to use a replaced the API which we have already built following the OpenAPI spec. So this is a node API built using Express in GitHub Codespaces.

First, let's simply explore the API. To do that, I've set the breakpoint in the project already and started the debug configuration, so let's switch over to the Swagger UI.

Like you can see, the repairs API supports basic create, update list and delete operations on the repairs entity. Let's explore this. Let's go ahead and look for all the repairs assigned to a maintenance staff member.

When I do that, the breakpoint is hit in my project, and like you can see, the response is returned, and the response contains basic metadata about the repairs entity itself, like the ID, the description, the staff member to whom it's assigned to, and so on.

That's it. This is a simple API which we are now ready to convert into a plugin for the copilot. To do that, we're going to switch over to the Teams Toolkit extension, which is right here. I'll say, "Create a new app," and then from the "new project" dropdown, I'm going to say, "Plugin for Copilot."

I'm prompted to enter an OpenAPI spec that defines my API shape and form. I already have one created for our repairs API, so I'll reference that. Like you can see now, I'm also prompted to select the methods in my API that I want to expose as operations in my plugin. I only want list and create repairs as the two operations that my plugin should support, so I'll select those. Of course I need a project folder location and a project name itself, and we are ready to go.

Now, the new project is being created and opened in this new browser tab. Now a plugin has two key components. No. 1 is the manifest file that defines the plugin's basic configuration metadata. And No. 2, are the adaptive cards which define the UI using which users will interact with my plugin.

Now, the cool thing here is using the OpenAPI spec which we just referenced a few seconds ago, the Toolkit extension has already created both the manifest file and adaptive cards for me automatically.

So let's click into each of them and explore. First, the manifest file. Like I said, the manifest defines the basic metadata and configuration about my plugin, like the plugin name, the description and all the operations we just explored exposed for our plugin.

Now, plugins can also support authentication, so if your plugin supports auth, you can specify that in the manifest. The repairs API supports anonymous access, so I'll skip that step.

Next, let's click into adaptive cards. Like I said, adaptive cards define the UI using which users will interact with my plugin. Using the repairs API response parameters, the Teams Toolkit extension has already created adaptive cards for me, which of course I can change.

For instance, I'd like for the thumbnail image of the repair ID, the repairs item to be a bit larger. So I'll go ahead and make the change. And like you just saw, the changes are reflected in real time in the preview surface, making it super easy for me to make these changes.

All right, so we made a couple of changes, and we are ready to test the plugin. To do that, we'll switch over to the debug configuration and then say, "Launch in Copilot." When I do that, Copilot will be opened in Microsoft Teams and my plugin will be sideloaded and made available for testing. So now let's – we are ready to run a quick test.

Now, Copilot will open in Microsoft Teams and I can go ahead and test with the prompt to fetch the repairs that are assigned to a maintenance staff member. When I do that, the repairs API that we just specified in the manifest is being invoked and the response from the API is being fetched and rendered in the form of both text and adaptive cards which we designed in the toolkit.

So there you have it.

(Applause.)

Awesome. So we just built a plugin using an existing API and have it already working in Copilot in Microsoft Teams. And we did all this with just a few steps in under five minutes. Go build this. Thank you.

(Applause.)

**RAJESH JHA:** Thank you, Archana, awesome.

So I think this toolkit to build a plugin is going to be a huge timesaver for the developers to extend the copilot capabilities. We would love your feedback as you use the tool in the coming weeks.

So we talked about extending it, you know, at the cross-application layer. I want to now talk about extending the copilot at the data tier.

So the Microsoft Graph, think of the Microsoft Graph as something that gets built and refreshed every single day from the trillions of signals that we get from customers as they use Microsoft 365.

Every day, we see hundreds of billions of documents and messages. And so what the Graph is, it's partitioned by customer, by tenant because the Graph belongs to the customer, and it is permissioned by the user. So whatever I have access to is what the Graph represents.

So we take the Microsoft Graph and that is the brain behind the copilot's amazing reasoning capability. So we take the Graph, we ground the AI, and all the user's context. And like I said, the security and the permissions are maintained because the Graph represents exactly my, the individual's permissions.

And so the Graph is being used, the customer's data is being used to ground the AI. Of course, we did not train the AI on the customer's data. We use that to ground it, and with the Microsoft enterprise compliance promise, basically we treat all the interaction of the user with the AI no different than we treat their documents or messages.

Now, there's one other very important thing that we are doing to bring the Graph forward for the world of natural language. Human language, natural language needs a deeper semantic understanding of the Graph, and so we are now building the semantic index. Let me talk about that for a bit.

The semantic index is of course, a vector index. It's an embedding space for the billions and billions of entities in the Graph. But it is much more than that. It is an understanding of the relationships that exist in the Graph.

What do I mean by relationship? Every one of us, we work with colleagues. There are some meetings that really matter to us, some documents that are trending in our organization, the projects we care about, the tasks assigned to us. That is a relationship that is captured in the semantic index. That is what allows us to then take the natural language intent expressed by the user and get the most relevant data to go into the AI. And that is what we want to make available to developers.

I want the semantic index to be as powerful an asset for developers as we use it internally for first-party. And the way you do that is by extending the semantic index with Graph connectors.

And to that end, I'm going to invite Yina, who's our expert. She's going to talk about how plugins and connectors work together. Welcome, Yina.

(Applause.)

**YINA ARENAS:** Thank you, Rajesh. Hello, Build. I am Yina, and I lead the Microsoft Graph team. So far, you have seen the power of plugins integrated with copilots and how you can get started building them. We have also shown the value of building a Microsoft Graph connector to bring your data into the semantic index for Copilot and message extensions to integrate seamlessly in Teams and Outlook.

Now, I'm going to show you a scenario that illustrates how all of these things can work in Microsoft 365 Copilot together.

In this demo, we're showing you how Daniella, who's a project manager at Dentsu, which is an integrated marketing and media company, might manage the process of updating the branding of a client's website.

First, she uses Copilot to gather and summarize information about the website deployment guidelines. Second, she uses Copilot to invoke plugins in context. Here, she's creating a ticket in her project management tool and assigning it to someone on her team. And finally, she asked Copilot to retrieve and update a Contoso logo to the newly created project management ticket.

Now, here, all of these happened on a snap, but consider how tedious it would have been to do all of these manually. It requires multiple steps, aggregating information from multiple sources and working across many different interfaces. With Copilot, she can do all of it without breaking the flow of work.

Now let's take a closer look at what is really going on, in each step on this process.



First, this is the first turn, it's gathering information. Here, you're seeing the deep retrieval and summarization capabilities of Copilot. Copilot is not only able to traverse all the data that Dentsu has in Microsoft Graph to include SharePoint documents, Outlook emails, Teams chats with colleagues and more, but it is also traversing data from Confluence, using a Confluence Microsoft Graph Connector, which is bringing index data from Confluence into Microsoft Graph.

In addition to what Copilot extracted from Confluence, it also includes new steps in the process that were identified from email data, like the rollback plan, updating the logo and the performance monitoring. Copilot returns a summary of all of the deployment guidelines with citation for each relevant artifact, chats, emails and documents. And now, because Copilot is powered by full semantic search, it was able to include steps in the deployment guidelines that otherwise might have been missed.

Now, the second turn. Here's where we are invoking plugins in context. Dentsu uses Jira as their ticket management tool. Jira integrates with Teams and Outlook using a message extension. With Copilot, the Jira message extension now functions as a plugin and creates a ticket using the context and information gathered in previous terms. It pre-populates several fields, like the project name and the summary, and it also includes a description based on the context it found on an email.

But as helpful as Copilot might be, we still need to make a few edits, like adding the ticket owner. And then we're all set.

Now, back in Teams, Copilot has already summarized all of the actions, as we're going to see in a second, including the reference to the newly created ticket.

Now, the third and final turn for this demo. Here's where Copilot interacts with an enterprise application, in this case to retrieve a branding asset. Dentsu's brand web catalog is their internally developed CS assets application, which integrates with Teams using message extensions. Using precise language in the prompt, like the example in the asset catalog, increases the fidelity of Copilot's response.

When we have confirmed that Copilot has retrieved the right branding asset from this line-of-business tool, it is just a matter of telling Copilot to update the ticket with the Contoso logo for reference, and Copilot does the rest. Finding the previously created ticket from the context, invoking the comments dialogs, and pre-populated the comments with a link to the logo file. Copilot then summarizes the changes to the Jira ticket and includes the references to the artifacts, and we're done. (Applause.)

To summarize, as a developer, you can use Microsoft Graph Connectors to bring your data into the semantic index for Copilot. You can also use plugins and build plugins to integrate skills into Copilot. These plugins can be from the OpenAI standard, or they can be message extensions. And finally, Copilot can combine insights from Microsoft Graph connected data with actions coming from the integrated plugins into multiturn processes, boosting productivity and keeping

customers in the flow of work, all while preserving the integrity of your app experiences and the attribution to its content.

Thank you very much. Back to you, Rajesh. (Applause.)

**RAJESH JHA:** Thank you, Yina. That was awesome. What we just saw was how, by using the Microsoft Graph, you get the full power of the semantic index that we built. And we want developers to be able to take full advantage of our investments in Microsoft Graph and the semantic index, so we have a deeper understanding of the user's intent.

Now let's talk about extending in-application workflows. Good news is the way you extend that is the exact same way you extended it across application capability. The same architecture for plugins and message extensions also work here, and this is about bringing the full power of all the community. You think about our Office applications in your Windows, there's so much richness. Now we're going to unlock that with natural language, and you're going to be able to extend that as developers.

And so, I'm going to invite now Wamwitha to actually show you this in action. Welcome, Wamwitha.

**WAMWITHA LOVE:** Thank you, Rajesh. (Applause.) Good morning, everyone.

In this demo, I'm Daniella, an account manager at Adatum, a renewable energy company, and I have an upcoming sales meeting with RellaCloud. Let's see how easy it is for me to prepare for my meeting using the Viva Sales integration with Copilot.

Starting in Word, I use Copilot with Viva Sales to quickly assemble several relevant resources from my CRM system, my calendar and Word documents from my company's OneDrive. Copilot then generates the summary document for me using the information gathered. And as I scan through the document, I notice a specific deal that looks relevant to my upcoming meeting. I prompt Copilot for more details.

Copilot uses Viva Sales to retrieve a detailed summary of the deal and prompts me to add it into the Word document. With the document updated, I save it to my CRM system.

I can also use Copilot with Viva Sales during my pitch call on Teams. Copilot can process my customer conversation in real time. Using Viva Sales, it's able to recognize that the customer has mentioned a competitor, ProsWare, is also pitching to RellaCloud. Viva Sales prompts me to get more information on ProsWare and quickly returns an analysis of its strengths and weaknesses.

And when the customer asks for insights into how similar companies have deployed Adatum technologies, I can prompt Copilot to pull up the customer success mural that my field engineers and I use to document our success stories and share them directly into the meeting. We agree on next steps, including a formal proposal review, and we're done.

Now, I want to create an engagement report, something that would normally take me about 30 minutes to complete, but without leaving Teams, I can prompt Copilot in Viva Sales to do it for me, using the meeting transcripts from my Teams call. Soon, Copilot produces a Word document I can edit and save into RellaCloud's CRM records.

After my conversation with RellaCloud, I think I'm close to closing the deal. I send an email to Kat on the legal team asking for help drafting a contract with a 15% discount applied. I've attached a typical master agreement for reference, and when Kat receives my email, Copilot gives her the option to draft the requested contract in Word. And when Kat launches Word, Copilot already has the context it needs to assemble a draft. All Kat needs to do is hit "generate."

Kat's team uses Thomson Reuters Legal Services. The Thomson Reuters plugin enables access to multiple trusted resources, including Practical Law to support drafting, and Westlaw to assist with legal research. Kat can modify the contract generated by Copilot for Word. For example, she can replace the limitation of liability clause and verify her changes are enforceable in California.

Kat prompts Copilot to provide her with a summary of changes by clause for a quick review, using Thomson Reuters Document Intelligence Service to quickly extract and compare modified clauses. With her review complete, Kat can now send the contract back to me for presentation to RellaCloud.

Thank you. Back to you, Rajesh. (Applause.)

**RAJESH JHA:** Thank you, Wamwitha. It's pretty cool to see how these sophisticated workflows are all happening in the application, and developers adding to just the friction-free nature of completing these business processes.

In a similar way, we've been working with partners like SAP, and there was this announcement last week of how SAP Success Factor is plugging into the copilot inside of Microsoft Word, using the copilot extensibility. And so, for example, if I'm creating a job description, the data in the SAP Success Factor helps you create this with all the right metadata, all the finetuning made possible by the business logic in SAP Success Factor.

In a similar way, I'm also very excited to announce the Microsoft Syntex plugin for the copilot. For those of you who may not know, Microsoft Syntex is content AI. Every organization has tons and tons of content, whether it be in SharePoint or OneDrive or Teams, and a bunch of other places.

And so, Microsoft Syntex is an AI tool that enables classification, that enables workflows that are possible on this content, unlocking the knowledge in an organization. And now, their entire corpus is available as a plugin in Copilot, and that allows business processes to be grounded in the user's context in the plugin. Let's take a look.

(Video segment.)

**VOICEOVER:** The Syntex plugin makes it easy to add new content as a column, automatically extracting metadata all from within Microsoft 365 Copilot. You can leverage the metadata to search and find relevant content with precision.

The plugin will enable you to build new documents from structured templates, combining data sources and your own inputs, and review that new document spotlighting key details. When you're done, the same plugin can add Syntex e-signatures to your documents for your approval, all from within Microsoft 365 Copilot.

(End video segment.)

**RAJESH JHA:** What we've seen so far, if I was to summarize the extensibility that we walked through, first, plugins. Whether these are ChatGPT-style plugins, all you need is an open API and a manifest, but there would be a message extension, your existing investments. And then we showed a toolkit, the Teams toolkit. And finally, we talked about how to extend the Graph with Graph Connectors.

But we know that what's really important is to enable your success as developers. And we've really been working on the feedback we've gotten from developers in terms of it's not just about the productivity of the developers, but it's about the impact of the work that you do. How do we get you distribution and reach? How do we enable end user discovery of the work that you do? And then, if you're an ISV, how do we help you monetize that?

Let me just take a few minutes and walk through each one of these.

On productivity, just as you saw in the demos today, we make it possible for you to create a plugin easily. You can customize the user experience with adaptive cards, and then you can debug and deploy this once and have it work in the host applications or as Copilot extensibility.

In terms of reach, of course, in the organization, IT has the accountability for making sure any application is secure, the data privacy is maintained. And you have to earn IT trust, and it's something that Microsoft has spent a lot of time with IT on. And so, our investments on reach really are to help developers to make their applications compliant.

What we have is a tool now that assesses and suggests remediation for your application to win it trust. Based on feedback from developers, we've created a Microsoft 365 Certification program. We take a look at the controls that industry standard frameworks recommend, and we assess your application against that. And if you get the Microsoft 365 Certification, that builds trust with IT and will speed up the deployment of your application.

And for IT, we have invested in different views, and badging and filtering so they can judge which application fits which role, what is the security profile of these apps. And the thing that we recently brought, which is a win/win/win, win for IT, win for end user and win for developers, is this auto installation of these applications and extensibility. If IT has trust in your application, and they want to deploy that to some set of users, they can auto-deploy it, so the user doesn't have to go, one by one, go find the application and then deploy it.

Reach is not enough. You have to get the end user to discover your application. Now, none of us go into work thinking, oh, let's install a new application. Usually, what happens here is we go by the recommendation from our colleagues, from our leaders, and of course, IT recommendations.

And so, what are we doing in the realm of discovery, of course, we first went and solidified – we were a little chaotic. We've got now a one-story experience in Office, and Teams and Outlook, where users can go discover applications and extensibility.

The thing that we've recently enabled that I'm very excited about is what we call seamless discovery with link unfurling. If somebody sends me a message and I don't have the application installed that the message refers to, we go to schema.org and we can show you the experience, even if the recipient doesn't have the application. This helps the virality of your apps.

And then, of course, we've made sure we've added user experiences to get the right applications to surface in the right context. For example, in Microsoft Teams, if you're in the meeting stage, you can go and figure out the relevant applications in meetings and bring that in. Same thing with channels and chat.

And in terms of monetization for ISVs, we've created a commercial marketplace. Now ISVs, you can figure out how to distribute your app at a regional level, at a department level, at a global level. And for IT, we've given them tools on how to manage users and licenses.

Taking a step back at what we've talked about today, and my call to action to all of our developer colleagues here, as we go forward together in this world of AI, go build Teams message extensions today, because these also act as Copilot extensibility. Take advantage of our investments in the semantic index and the Microsoft Graph by annotating the Graph with your data and metadata, and then give us feedback on the Teams plugin, which is really about streamlining your ability to build, debug and deploy plugins.

Now, we have several other exciting announcements beyond Copilot. I'm sure you've heard the word "copilot" multiple times today, but we have other announcements. And so, I really want to encourage you to take a look at the breakouts in the demos for some other very exciting announcements. But there are two things I want to spend a few more minutes on. One is Teams Live Share, and the other is Microsoft Mesh.

The Teams Live Share, what you see here is Autodesk, multiplayer Autodesk. This is an SDK that is now released, is generally available. This SDK is about bringing gaming quality multiuser capability to your application. You don't have to do any backend work. We've done the work. We use this ourselves for Excel Live and PowerPoint Live.

The idea here is in a hybrid work, in a distributed workforce across time zones, across geographies, don't you want your applications when they are brought in the context of a meeting to be co-created, co-edited, co-annotated? And that's what this Teams Live SDK does. This is now generally available. (Applause.)

Let's talk about Microsoft Mesh. This is about building connections. It's about building fun, about immersive experiences. Now you can create personalized avatars for Microsoft Teams. Your reactions, your personality is represented in the meeting even when the video is on because of video fatigue is a real thing.

I'm also very excited that in private preview, we have two new capabilities. As developers, you can extend any Teams meeting to be an immersive experience to build stronger connection. And it doesn't matter; the cool thing is it doesn't matter if one of the attendees has a headset on and in 3D, and the others are in 2D. It works in this heterogeneous set up, and really, you can build now any meeting. You can build templates for immersive meetings.

And the same technology is now available for you to build custom immersive experiences. Imagine there's a town hall or employee onboarding or some social gathering. It's all made possible about Microsoft Mesh. Let's take a look.

(Video segment.)

**VOICEOVER:** Sometimes hybrid meetings can feel a little impersonal. Avatars for Microsoft Teams gives people the choice to show up how they want, helping them feel more comfortable and included.

But can we take digital presence a step further? Immersive spaces for Teams help the interactions feel more engaging, more connected and more natural.

But what if we could take it even further by building a custom immersive experience just for your organization? Microsoft Mesh elevates orientations, trainings and employee experiences by transcending time and space.

No matter where you are, you can come together in one place, connect like never before with Microsoft Teams and Microsoft Mesh.

(End video segment.) (Applause.)

**RAJESH JHA:** We've talked a lot about plugins. I talked about how you can build a plugin, have it work in ChatGPT in Bing and Microsoft 365. There's one more place that your plugins will reach users, and that place has a billion users, and that is Windows. And to tell you more about that opportunity with Windows, let me invite Panos to the stage. (Applause.)

END