

05222024 Build Keynote Scott Guthrie

Microsoft Build 2024
Scott Guthrie, Charles Lamanna, Eric Boyd
Redmond, Washington
May 22, 2024

SCOTT GUTHRIE: Well good morning everyone. It's great to be back here at Microsoft Build. AI is transforming the world and developers are at the very center of this. It's never been a better time to be a developer. AI is going to profoundly change how we work and how every organization operates. Every existing app is going to be reinvented with AI, and we're going to see new apps built using AI that weren't possible before.

Yesterday, Satya, Kevin and Rajesh covered Microsoft Copilot and touched on what's new across the Copilot stack. Today I'm going to walk through the Copilot stack, which is the most advanced platform for creating AI capabilities and solutions. We're going to show a lot of code along the way and introduce you to some amazing new capabilities that we're shipping this week at Build.

Let's start by talking about our developer tools. The Visual Studio family of products, which includes both Visual Studio and Visual Studio Code, are now the most widely used development tools in the world, with more than 40 million active developers. Copilot Studio, which we released last year, is now used by more than 30,000 organizations to build AI solutions, and GitHub is used by over 100 million developers, and is literally the home of open source.

GitHub provides developers and enterprises an integrated, AI powered, end-to-end developer platform that can be used to collaborate, automate and secure DevOps solutions. One of the most groundbreaking capabilities that we've launched with GitHub is GitHub Copilot.

GitHub Copilot was the first Copilot solution that we built here at Microsoft, using the transformational large AI models, and it provides an AI paired programmer that works with all popular programming languages and dramatically accelerates developer productivity. Millions of developers now use GitHub Copilot daily.

Now, a few weeks ago, we launched the preview of GitHub Copilot Workspace, which takes GitHub Copilot to the next level, and we're getting phenomenal feedback from developers using it. With GitHub Copilot Workspace, you can now brainstorm, plan, build and test code all using natural language. It's designed to reduce complexity and improve productivity without taking away the aspects of software development that you value most, such as decision making, creativity and ownership. Developers always remain in full control over every step of the process.

Let's watch a demo of GitHub Copilot Workspace in action.

VIDEO SEGMENT: GitHub Copilot is a mature AI paired programmer used by millions of developers. It helps you write your code faster and more efficiently, predicting what you need by getting context from within your favorite code editor, saving you time and keystrokes.

With Copilot Chat in line, you can also talk to Copilot right in the editor. For example, it's great at helping you document the code you're working on, even taking into account the practices and standards of the language you're working in. But Copilot Chat can also help developers get up to speed quickly when working with unfamiliar code.

Imagine you've been tasked with changing the landing page on an app that you're completely new to. It can be overwhelming to know where to start. GitHub Copilot Chat can now look at the entire project to give you answers, referencing much more than just the open file. Here, it saves me time by identifying the file I need to look at to get started on my work. Of course, from there, I still need to understand what the code does so I can just ask.

Developers don't just write code. There's a lot of work in figuring out what needs to change and deciding on a plan of action. In GitHub, developers usually start with an issue like this one, adding the price of our products to the home page. GitHub Copilot can start here too with GitHub Copilot Workspace.

Copilot Workspace can use the details in your issue and its knowledge of your code base to build a specification determining the relevant current state of your code and a desired end state. This is all fully iterative. You can change every step in the proposed spec if your issue didn't fully capture everything that needed to be done. This one looks great.

From here, Copilot can generate a plan with details of the files that need to change and what those changes are. Then you can ask Copilot Workspace to implement those changes. You can even see a live preview to make sure you've got the result you need. Hmm, that's good, but I think the price needs to stand out a little more.

Luckily, you're in control every step of the way. You can keep iterating by editing the plan to help guide Copilot Workspace to the solution you're happy with, and so let's do that and get Copilot to update that file in Workspace. And you're still in control. After the code has been generated, you can edit it right in Workspace.

For the moment, let's make one more small change, and perfect. I think I'm done. I can now submit a pull request right from here, ready for a review by my team. Copilot even generates a helpful description of the changes I just made. AI can also help keep our code secure.

Let's look at another pull request, and straight away I can see there's a problem. These changes introduce a security vulnerability that I honestly might have missed. In my review, GitHub Advanced Security was able to analyze the code and identify the vulnerability, and now with the help of AI, it can even suggest a fix for the issue. You can merge that fix with one click or debug and test the changes in an editor if you want to examine them more closely.

More than ever, GitHub is the AI-powered developer platform, making you more productive wherever you're working.

SCOTT GUTHRIE: So over 50,000 organizations are already using GitHub Copilot Business today to supercharge the productivity of their developers. At Airbnb, approximately 70% of developers are using GitHub Copilot on a weekly basis. Shopify estimates they accept up to 25,000 lines of code every day, again using GitHub Copilot. And in a survey, nearly half of global car developers estimated that they saved between one to three hours a week using GitHub Copilot.

Let's hear from developers from several other organizations about how they're using GitHub Copilot and how it's transforming their engineering teams.

VIDEO SEGMENT: When I think about our engineers, the thing I want them to be spending time on is how to make our users' lives easier, and that's where Copilot really shines.

I love GitHub Copilot. Once it's in your IDE, it just feels natural and then trying to code without it feels really strange.

AI can get really overwhelming. What I loved about Copilot is that it feels transparent.

Gen AI means that we can go from spending hours, maybe searching on Stack Overflow, searching on Google for how to do something, and we can ask our Copilot how to do it instead, but that's not where the story ends. Once you've written your code, you need to be able to test it. You need other people to be able to understand what your code is doing. Rather than thinking, how I can test the code, I can ask Copilot "How am I going to test this code?" I can ask Copilot.

Isn't that the dream to have a Copilot assistant that just helps you and gives you knowledge about the code base, making it easy to explain what's going on in a specific file, like a no-brainer.

SCOTT GUTHRIE: Azure provides a rich and flexible set of application platform services that you can use to run any application using any programming language. These are the same services that are running Microsoft Teams, Microsoft 365, Dynamics 365 and GitHub Copilot, all at scale securely for hundreds of millions of users around the world, and they're powering our customers' most sophisticated AI apps. With Azure Application Services, you can focus on building great apps, knowing that they're going to deliver the performance and the scale that you need.

Now, millions of service as solutions are now powered by our Azure App platform, and increasingly, these apps are deeply integrating AI. The Australian supermarket Coles, for example, generates over 1.6 billion daily AI recommendations, dynamically updating their experience to give customers the best digital experience both in-store and online.

Dick's Sporting Goods is delivering a one-store digital experience using AI, making every customer interaction personal and engaging, using apps running on top of Azure Kubernetes Service.

This week at Build, we're introducing our new AKS Automatic Offer. We're taking years of experience running some of the largest and most advanced Kubernetes applications in the world and are embedding this knowledge as best practices within AKS Automatic.

AKS automates everything from cluster setup and management to performance and security. It's production ready by default, and it's preconfigured and optimized to ensure the highest performance for your application. It has robust deployment safeguards and security policies, all built in.

And so as a developer, you now have access to a self-service app platform that moves you from code to deployed Kubernetes apps in minutes while providing the full power of the Kubernetes API and ecosystem.

I'm also really excited to announce that .NET Aspire is now generally available. It's open sourced and designed to make cloud native development simple and enable developers to be more productive.

.NET aspire is going to make it much easier for developers to build solutions and do it in a repeatable, safe, secure and reliable way, taking full advantage of Visual Studio .NET and GitHub.

Now, the combination of GitHub and Azure is incredibly powerful and provides developers with an end-to-end experience that is optimized for the era of AI. We are working to remove the friction across the entire development cloud lifecycle so that developers can really focus on building and running great apps.

With GitHub Copilot, we've also been trying to make the entire process more efficient and more intelligent. With our new GitHub Copilot for Azure support, you can now use natural language to enable an even tighter developer loop with Azure.

Developers can ask questions not just about their code, but also about their Azure cloud environment and all the resources that they're working with. It enables developers to stay in the flow inside their editing environment and take advantage of the ultimate AI powered development and cloud platform combination.

Let's watch a video showing GitHub Copilot for Azure in action.

VIDEO SEGMENT: Let's take a look at how GitHub Copilot for Azure can help both experienced developers and developers who are new to Azure get started quickly and confidently. GitHub Copilot for Azure is an extension for GitHub Copilot that helps you learn about Azure, manage Azure resources, and troubleshoot issues directly within Visual Studio Code and Visual Studio.

I'm working on a new project that uses Azure OpenAI, so I need to find out what Azure OpenAI resources I have access to. I can use GitHub Copilot for Azure to help me with this question.

For help with Azure specific topics, I start my question to GitHub Copilot in the chat window with @Azure, and I can ask, "What Azure OpenAI resources do I have deployed?" I can quickly see information about my existing resources and the regions they are located in. I can also use @Azure to help me quickly find details about my Azure resources, such as the default domain for a web app.

GitHub Copilot for Azure can help guide me on how to deploy my applications. I know that I want to deploy this application to Kubernetes, but I'm new to using Azure Kubernetes Service.

For this project, Copilot for Azure recommends AKS Automatic as the easiest way to get started with Kubernetes on Azure and walks the user through commands to create a cluster using AKS Automatic and deploy the application.

Just like with GitHub Copilot, the user is always in control of the actions that are being taken.

First, the agent guides me to initialize my project, which will configure and add the infrastructure-as-code files needed to deploy this application to AKS Automatic. I'll confirm in the terminal that I do want to add these files.

Initializing my project also created a GitHub action to deploy the service as part of my CICD process, and so now, any time another developer on my team commits a change, the action will be triggered and changes will be automatically deployed to my new AKS cluster.

Our most recent change has triggered the action to run. Using the VS Code GitHub actions extension, I can quickly check the status of my action directly in VS Code, and I can easily navigate to view the action in GitHub to see more details.

After this action is complete, I can continue iterating in VS Code and GitHub. I can use the VS Code extensions for working with cloud native tools like Kubernetes and Docker. And if I need to explore Azure to learn more about my deployed application, the extensions make it easy for me to find exactly what I need.

For example, if I want to view the resources that were deployed by the GitHub action, it's a single click from VS Code to the specific location in the portal that provides the information I'm looking for.

In just a few minutes, GitHub Copilot for Azure has helped me deploy my application to Kubernetes in Azure. All from within VS Code.

SCOTT GUTHRIE: With Visual Studio in GitHub, we have the most widely used developer tools. With Microsoft Copilot Studio, we're providing you with a tool that enables you to build

Copilot solutions even more quickly. Copilot Studio can be used both standalone as well as together with Visual Studio and GitHub.

To share more about Microsoft Copilot Studio I'd like to invite Charles Lamanna, who leads our Business Application and Platform team to the stage to show it off. Please welcome Charles.

CHARLES LAMANNA: Thank you, Scott.

Last fall, we introduced Copilot Studio to enable you to do two things: easily extend Microsoft Copilot and create your own Copilot. These Copilots can be applied to many internal processes to streamline employee operations, things like HR compliance, or to go create a Copilot to better engage your customers. Best of all, these Copilots can be published anywhere you want that's inside Microsoft 365 or Teams or inside of your web and mobile applications.

This year at Build, we have some exciting Copilot Studio announcements. Copilots evolved from working alongside you to also working for you. What that means is proactively and independently, the new agent capabilities make it so Copilot Studio can unlock your Copilot to automate tasks asynchronously and in the background.

Copilot is so much more than a chat bot now, and you can provide your Copilot with a defined task equipped with public and enterprise data sources, and then Copilot will orchestrate dynamic workflows to automate them end-to-end.

This is best seen in a live demo, so let me show you.

Over here, we are looking at Fabrikam, which is a fictitious telecommunications company, Fabrikam, that helps its customers, which are businesses and consumers, get new phones and change their plans all the time. And to do that, they have a great website, but they wanted to go beyond a website and have a Copilot for their customers.

So what they've done is they've embedded a great customer Copilot right inside here. And what I can see is I have a path where I can say if I'm a new or an existing customer, and I'm going to go say I'm a new customer, and in this case, it's offering to compare my current plan with what Fabrikam can offer me.

What it tells me is that I can upload my last bill for my current carrier and the Copilot will read through the bill the plans that Fabrikam offers, and then let me know what I can switch to save the most money and even improve my experience. And I happen to have a bill handy from Contoso phone carrier.

And if we look here, I have three lines, I have a data plan, and I end up paying about \$232 per month, but Fabrikam can definitely do better than that. I come back over here, upload that file I just showed you, let the Copilot know I've uploaded everything that I want, and it will come back and give me a recommendation.

What's great here is it's been able to look through that bill, and it knows that I pay \$230 per month. It's been able to look through Fabrikam's internal documents to know I can get a plan for \$150 per month, which means I can save almost \$1,000 a year.

This type of conversational, natural experience is something that you can really only do with a Copilot, and that seems like it will take a lot of work, but I'll show how you can build all of that inside of Copilot Studio.

So I switch over here inside Copilot Studio, and we have a bunch of templates available that you can get started with, or you can even use natural language to describe what you want your Copilot to do, and it will start laying out the framework and scaffolding. I only have five more minutes, so I'm just going to jump to one that I've pre-built before.

You can see this is that customer Copilot for Fabrikam inside of Copilot Studio. What you do to define these Copilots is you provide instructions which are like the meta prompts which are the guardrails and guidelines for your Copilot.

You're then able to go configure your knowledge sources, which is what Copilot uses for context and information, and things called topics, which I'll cover in a second, and then actions where you can enable your Copilot to go run workflows or do things in the background.

And then what happens is we use the large language model in the orchestrator to self-assemble these actions, this knowledge, these topics, these instructions to answer any question that a customer comes in with, which is why the Copilot is so capable and you can build it very quickly.

Now let's go look at that knowledge in more detail.

In my case for the demo, I already have a few different knowledge sources registered. I have my public website, I have some SharePoint sites, I have a PDF I've uploaded, and of course I use Dataverse on the back end. All of this is automatically having its embeddings generated, stored in a vector database that supports semantic search, semantic query, and that just happens for me for free. I don't have to worry about those details and I can go bring in other knowledge sources.

So if I want more than those ones, I can use Microsoft Fabric, Azure Data Service Services, SAP, ServiceNow, you name it, and you can bring those into your Copilot. This is what Copilot uses to answer those questions and figure out how to execute over time.

Now, I don't just want my Copilot to be about Q&A. I also want my Copilot to be able to take actions. You can see here I already have a bunch of registered actions like payment gateway connector where I can actually generate links for customers to pay, or I can look up customer history and more. It's a really rich set of functions that your Copilot can use.

What's great is we have over 1,400 connectors available in Copilot Studio to systems in the cloud and on-premise, so you can get access and run actions and workflows no matter where

your data is. What's great about that is, if we don't have a connector, you also have the ability to register Azure API Endpoints.

What I've done is I want I want to use this Azure API for device trade-in, and you'll see this in a little bit in Seth's demo later in the keynote, where this API will be invoked by the Copilot to help a customer trade in their phone for a new one. And if there's ever anything you want to bring, you can go use any of those Azure application services Scott was talking about right inside of Copilot Studio.

Now, knowledge and actions are great, but they're a little non-deterministic. What we hear from customers is that first, 80% of your Copilot is really easy to build as a result, but there is that last 20%, the super-important workloads you want to control, and for that you can use Topics.

In the case of, let's say, a device trade-in, I want to have exact control of the dialogue flow and the conversation with my customer. We can see that this topic will only trigger for any of these dialogs or queries from the customer. That way I can run a very specific workflow, which maybe calls APIs and gives particular dialogs based on what the user is trying to do for that device trade-in.

This ability to have kind of generalized orchestration and self-assembly and overrides in Topics makes your Copilot super-powerful, super-quickly without having to compromise on the overall experience. And once I've configured all of that, I can then go publish to tons of different channels.

In my case, I've used a custom website of course, to embed it for Fabrikam, but you can use all kinds of other digital channels, and before you even publish it, you can also easily test right inside of Copilot Studio.

So if I look over here, this is that same conversational experience I saw on my website, and I can simulate before I go live. All of this comes together for an incredibly robust Copilot experience, but what we've really internalized over the last year is you don't just want your Copilot to chat or have a conversation. You also want your Copilot to be able to run in the background to run asynchronously and execute workflows. That's what makes a Copilot have agent capabilities.

So what we're excited to announce at Build this year is you can now register triggers for your Copilot. Triggers allow data events from a database or an application to start the execution of your Copilot. What happens is, instead of a conversation being the prompt, it's the data from the trigger that becomes the prompt for the Copilot. These can run in the background, which is a little tricky to show on stage, and so I'm going to go to one that ran earlier.

You can see here, I can track all the steps that my Copilot took in the background, and so if a user requests a change to Fabrikam, I can do things like validate their user identity or look up and send the discounts. I can understand why the Copilot took that action. I can see the thought process and the reasoning that went into those steps.

So you can build these incredibly powerful Copilots. They can run talking to a customer or in the background, and you can understand what's driving its behavior. And this is what is important on day 100 or day 1,000 when you're trying to optimize your Copilot more and more. This is a great set of capabilities for Fabrikam, and this was a fictitious example Fabrikam, but many other customers have used Copilot Studio.

In fact, over 30,000 organizations have already used Copilot Studio, and it's been growing 175% quarter over quarter. PG&E is one of the largest gas and electric companies in the U.S., and they built a Copilot, which manages nearly 40% of calls for their IT helpdesk in Copilot Studio. Cineplex, a leading media entertainment company in Canada, built a Copilot for customer service agents, which reduced their handling time for customer queries from 15 minutes to 30 seconds.

These are just a few examples. We're incredibly excited to see what you build with Copilot Studio. And thank you so much for the time today. Back to you, Scott.

SCOTT GUTHRIE: Thanks, Charles.

Now that we've covered our developer tools and application services in Azure, let's talk about Azure AI. Customer adoption of Azure AI is accelerating, and we're already helping more than 50,000 companies around the globe achieve real business impact using it, and UiPath is one of those organizations. Their portfolio of AI powered solutions is helping transform the way people work with enterprise automation. Powered by our Azure OpenAI service, UiPath's communication mining solution has saved one insurance customer more than 90,000 work hours already.

And TomTom is bringing intelligence into the cars we drive with its new car infotainment system. Our response times to drivers using the new system improved by over 80% in just a few months, and the system is now able to understand and answer 95% of complex driver requests. Microsoft itself also runs on Azure AI. All the Microsoft Copilots are built on top of the same Azure AI platform that's available to all of you. This means you get the benefit of the battle hardening we've done to support tens of millions of users and organizations with AI around the world.

Now, you heard about innovations in Azure AI from Satya yesterday and the vision of our AI platform from Kevin and our partner at OpenAI, Sam Altman. Now, to share more about these announcements, I'd like to bring Eric Boyd, who leads the Azure AI engineering group, to the stage. Please welcome Eric.

ERIC BOYD: Thank you, Scott. No matter what your use case is, Azure AI has you covered. Our AI portfolio is housed in the Azure AI studio, which is a development hub for building generative AI solutions. Within the studio, you'll find services like Azure OpenAI service and Azure AI search, giving you access to rich models and services directly so you can get into production faster. We also offer Azure Machine Learning for advanced data science, custom model development and full model lifecycle management. Through our vast model catalog, you'll gain access to the latest cutting-edge models, as well as models as a service payment

option, which we'll cover in a bit. All of this is built on Azure's powerful supercomputing systems, with everything needed to run the most demanding, complex AI projects. It's even used by OpenAI itself.

Now, Scott mentioned Azure OpenAI services is one of our flagship products. We have more than 50,000 customers using Azure OpenAI, and more than one-third of them are new to Azure. They are turning to us as the trusted platform they can depend on. Microsoft is the only company that has more than a decade of leadership in security and compliance with Azure, and nearly limitless throughput backed by a resilient infrastructure, all backed by the best SLA in market. The last week, OpenAI unveiled GPT-4o, which is now the leading foundation model operating in English and other languages in text, audio and image processing. I'm pleased to announce the model is generally available with regional PAYG tokens and provisioned throughput on our Azure OpenAI service.

I'm also pleased to announce the public preview of our newest global PAYG token service, providing you with up to 10 million tokens per minute of throughput with this model. But that's not all. For the first time, we're announcing batch off-peak capacity at a 50% discount for your Azure OpenAI service to meet your unique needs. Your batch job will be processed within 24 hours, taking advantage of the capacity that frees up each night and saving you money as a result. Now, this is the same capacity pool that Microsoft runs on, and we're excited to provide you more availability to you at affordable prices. In fact, we're the first and only hyperscaler to make this capability available. We're also bringing a comprehensive set of Frontier and open-source models, enabling customers to choose the AI models for the right job, at the right cost and at the right quality. Together with our great open-source partnership with Hugging Face, Azure AI offers more than 1,600 foundational models from OpenAI, Microsoft Research, Meta, Mistral AI, Nvidia and even more.

Now, as developers go from experimentation to development, they've seen value in tuning the cost-performance curve for certain use cases, and the Phi family of models is a great option for this. Earlier this month, we introduced the most capable small language models, which operate at a cost-effective price point. These models punch a whole weight class above their size, but with the power of our built-in responsible AI. This week we are announcing Phi-3v, for vision, powering rich multimodal use cases. This is especially great for resource-constrained environments, including on-device, Edge, offline inferencing, latency bound scenarios where fast response time is critical.

Now, beyond our first-party models, I'm also pleased to announce the release of several new models as a service in Azure AI studio, including the leading Arabic language model, CORE42 of Jais, and Nixtla's TimeGen-1, with even more coming soon. Now, as AI organizations moved beyond model discovery, they are now deploying to production. The most common application pattern is the RAG application, and there the retrieval system is a critical component to ensure production level scalability and high-quality responses. Azure AI search supports massive customer applications, including all of OpenAI's GPTs, and is used by over half of the Fortune 500. We are making it easier for all of our customers to scale without compromising on cost or performance. And now for newly created search services, customers can scale in Azure AI Search Vector Index up to 12 times larger than before, at no additional cost. Now, Sam and

Kevin talked yesterday about some of the things that OpenAI has been doing, but let's look at how they're harnessing the power of Azure AI search and Azure Cosmos DB.

(Start video segment.)

OPENAI DEMO: At OpenAI, our mission is to make artificial intelligence both accessible and safe to all of humanity. ChatGPT has been the fastest-growing consumer app in all of history, with over 100 million weekly active users. Our API is trusted by over 2 million developers worldwide. We launched two RAG powered features. GPTs lets users build custom versions of ChatGPT that have external knowledge. Over 3 million GPTs have been built on the platform. That's a significant amount of retrieval augmented generation, or RAG, that is happening daily. The Assistants API lets developers build their own AI powered assistants to integrate external knowledge using RAG at scale.

To successfully support our massive user base, we need to use the best technology available; that's why we decided to use Azure AI Search and Azure Cosmos DB. Azure AI search is more than just a vector database, it is a retrieval system with hybrid search capabilities and support for metadata filtering. Our API users can now upload 500 times the number of files that they could earlier, and that's powered by Azure AI search. Azure Cosmos DB auto scales to meet our capacity needs while being enterprise grade and feature set in security. The Azure Stack has been very reliable and has worked really well to support our scale needs thus far. OpenAI and Microsoft both deeply believe in responsible AI. With Azure and our tools, we can solve some of the biggest problems of today and tomorrow.

(End video segment.)

ERIC BOYD: Now, six months ago, we announced the public preview of the Azure AI Studio, a unified platform designed for developers to build, manage and deploy applications moving from prototyping to scale quickly. I'm excited to announce that that's now generally available, with lots of users already signed up. To show you how you can use the Azure AI Studio to build your application, please welcome Seth Juarez, principal program manager for Azure AI, to the stage. Welcome, Seth. Back from camping already?

SETH JUAREZ: Back from camping.

ERIC BOYD: How'd those shoes work out for you?

SETH JUAREZ: My version of camping nowadays is pulling the blanket just a little bit more over my head at night. Having said that though, trading in my device is no joke because I want to do it as soon as possible. And as you saw earlier here, Charles built an amazing application that allowed customers to trade up into this particular Fabrikam instead of Contoso. It's a heated rivalry, trust me. Notice that there's some account stuff that you can do, but what I want to do is I want to be able to add a trade-in option so that you can go and add your phone and see if you can get it traded up.

Here you can see the back-end system, and what I'm going to do is I'm going to show you what that looks like (INAUDIBLE). I'm going to hit "trade-in device." What you're going to see is you're going to see my ability to upload my phone, I am going to control + shift + 1 here, and I'm going to get this upgraded. What it's going to do is it's going to call this connector, and this connector has an API that is being called. In fact, there's the answer. Here is the actual API. This is the connector. It's basically an API that you call, and you can see the answer is pretty good here. Let me zoom in here. Pretty good. Notice that it's able to look at the actual phone and tell you what is actually going on with it, which is super cool.

Now, the cool thing about this is that the reason why I want to actually build it is because our policies are internal, and we need to use our internal database systems and our internal things to make this part of Copilot. The way you do this is really a three-step process. The first step is to discover the AI capabilities you want to use, the second step is to develop with those capabilities, and the last step is to deliver these capabilities in a safe, reliable and observable way. You want to know what's actually going on. Let me show you how you do that. To get started, you will use Azure AI Studio. It is your one-stop shop for all things AI. In fact, when I say one-stop shop, I literally mean it. We have a lot of models. A lot of models. You can see them right here. This many. That's a lot. I almost wish it said like "l337" or something, but it's not, it's 1667. The reality is that we have a ton of models that do a lot of amazing things. In fact, GPT-4 Omni is the model that I want to use because it can look at images and it can process them.

But if you don't know what model you want to use, that's totally OK, too. You can go to model benchmarks, and you can compare all the models against various different data sets, against various different metrics. You can easily discover the models that you want to have. Azure AI Studio has a complete set of rich models to do amazing things to delight your customers. OK, that's step No. 1, discover. Step No. 2 is you start development. I am going to handcraft a prompt. Pretend I typed this. I'm typing this here really quickly. Notice that this is a prompt for Fabrikam mobile, and I'm kind of hard coding things in. Coding things into the prompt with my name.

Eventually we want to swap this out with database calls, etc., but let me show you this actually really does exactly the same thing that we want. I'm going to add this picture here. I am going to type in, "can I get this upgraded?" And you're going to see a very similar-ish response that feels natural but still has the actual right information. Notice it's similar, it feels more natural, but the information is still the same, which is super important. This is generally where things stop. What do I do now as a developer to make this go into my app? What we're excited to announce today is that there is a brand-new format that we're using called "promptTY". I like the name; it's so cute.

It's basically a way to move your prompt in a tiny little package down into your IDE in a reusable way. Let me show you what that looks like. I'm going to go over here to Visual Studio Code, my preferred environment. This is a tiny promptTY file. Notice it's exactly the same thing you saw before, but the difference is that you can plug some values in. If I scroll up, you're probably wondering where those values come from. When I'm quick testing, notice I can put some sample values in there. Let me quickly test it here and run it using the Visual Studio Code

extension. The reason why it knows that it can run it is because part of this format has the model and how you run it. Notice it says, "hi, Seth," but let's change this to something else.

Obviously, I'm on stage here presenting something, so I need to rename myself to this and let's see what's going on here. I'll scroll and you'll see that there's Scott. Notice that now you've actually moved the playground into your IDE, into a versionable asset that you can reuse, you can save, and you can use on all parts of your environment. You're probably wondering, what does the code look like for this? Let me go over here and show it to you. This is a Flask application. There is literally the API trade thing here, and there's one work function. This is the function that does all the work. I'm going to go to that function. There it is. I want to show you this is the extent of the entire application. This is all of it. Notice that we have the database calls to get the actual customer, we have the database calls to get the policy, and we literally just put those values into this file. That's pretty cool. The coolest part is that this file that I just changed, you can see now becomes part of your code base. It can be checked in and everything can happen. That is how you develop. That's the second step.

The third step is how do you deliver this in a safe, reliable and repeatable way? Well, I always use GitHub Actions. GitHub Actions is an amazing way to create repeatable processes, and one of the processes that I want to do is I want to make sure that this thing is evaluated every time we check things in. Let me go back to GitHub Actions here, and you are going to see these evaluations that run every single time I check in. When I go to these evaluations, you're going to see each one of the steps. One of them is groundedness. Groundedness tests how grounded the response is in the facts that we fetch from the database, like the policy and the customer information. Notice I got a five out of five, so we're able to deploy. There was one little sneaky thing that I forgot to tell you about this little trace thing, because in production, we want to make sure we observe that things are going correctly as well.

This little trace thing creates open telemetry Spans, which is an open-source observability tool. And because it's open source, of course it's going to work inside of App Insights. I've actually have App Insights open right here. Let me go to the transaction search and let me hit refresh. What you're going to see is the call I just made. That's the actual call. Then when I click on this and show the timeline, you're going to see the actual function that we just called. In production, you're able to see how these things are behaving. This is awesome. Again, in a few steps I showed you how to add -- yes, please. Thank you. In a few steps I showed you how to discover the model you want to use, develop using this new prompt format, and finally deliver in a safe, repeatable, and observable way. To tell us a little bit more about how to do this responsibly, please welcome Sara Bird, chief product officer for safety and reliability.

SARAH BIRD: Thanks, Seth. As you've seen over the past two days, we're at the cutting edge of AI innovation at scale, which means it's critical that we're also at the cutting edge of building responsibly. Everything starts with our AI principles. We're all learning so much on this journey, and at Microsoft, we've been learning for the past eight years while putting our principles into practice in our products. To do this, we infuse safety and security in every single layer of the stack, so our entire system is safe and secure by design. Let's see what this looks like under the hood. In a Copilot, the user prompt is combined with relevant data to help the system provide high-quality responses that are fresh and accurate. The safety system scans all of this to look for

problematic inputs, such as prompt injection attacks, to prevent them from ever reaching the model, and the model is trained with built-in safety mechanisms, so it knows how to respond appropriately.

However, like all AI systems, it does sometimes make mistakes, so the safety system also reviews the output to prevent the system from producing things like copyright materials or harmful content. That way, the user gets a high quality and safe response. We then monitor the system to ensure it's working effectively and to shut down ongoing attacks. We've built a lot of this into the Azure AI platform by default, so you don't have to worry about it. And because every application is unique, we also provide you with tools to customize and manage your safety experience. Let me show you how it works in the AI application that Seth just showed you.

(Start video segment.)

VIDEO DEMO: First, I'm going to create a custom content filter with Azure AI Content Safety to provide real-time protection on inputs and outputs aligned with my application's needs. For inputs, the default settings for harmful content look good to me. I'm also going to turn on Prompt Shield to protect against direct prompt injection attacks, which are attacks sent through the user prompt; and indirect attacks, which are attacks hidden in the grounding data. For outputs, I'm going to add protections for blocking copyright materials and ungrounded responses, which are where the model has inserted something not in the data. Now I can create and deploy.

Now that I have everything configured properly for my application, I'm going to evaluate the quality and safety of it overall. I can see that the application is getting good scores for the quality metrics of relevance, coherence and fluency, and it's staying grounded. This is helpful, but I also want to assess my application for safety metrics. I'm going to create a new evaluation called a "risk and safety check." One of the challenges with running safety evaluations is often people don't have high-quality data to test for certain risks, like having access to all the latest jailbreak techniques. However, we manage this for you with adversarial data set generation. I'm going to turn that on and then increase the sample count to 1,000, because I want to test at scale before deploying. I'm also going to increase the turn count to 10 to simulate longer conversations and tell the system I want to include testing with jailbreak style attacks.

I'm going to set it to test for all the categories of harmful content. This is going to take a while to run, so let's look at one that's already completed. Looks like the application is doing pretty well. About 4% of violent and sexual content is getting through, and 2% of the other categories. Now, this evaluation is not using jailbreak techniques, so let's compare how well my application does with those. It looks like all of the defect rates have gone up, and overall, 10% of jailbreak attacks are getting through. These rates look reasonable for adversarial tests, so we can go ahead and deploy. Now, let's try this in action in a SaaS application. I've turned on debug mode so we can see what happens behind the scenes. I'm going to try a jailbreak to see if I can get the system to give me credit card information.

As you can see, Azure AI Content Safety identifies the attack and deflects it, which is great. However, I don't know if this is someone playing around or potentially a real threat, so I want my SecOps team to take a look at it. Microsoft Defender for cloud is directly integrated with

Azure AI, so this attempt triggered an alert here in the Defender interface. I can see what the incident is. In this case, a prompt-induced credential theft attempt. I can see when it started, where it came from, and if it's still ongoing. Defender also tells me that the attempt was successfully deflected by Prompt Shield, so I know my system remains secure and I'm ready for whatever comes next.

(End video segment.)

SARAH BIRD: Azure AI Content Safety is a state-of-the-art safety system that detects issues in real time to prevent undesirable AI system behaviors. We've recently added new capabilities to address the most common risks we see. Prompt injection attacks and hallucination errors are protected against with Prompt Shields and groundedness detection. We've created risk and safety evaluations so you can test your application with AI-assisted adversarial inputs. This week at Build, we're excited to introduce custom categories, which allows you to quickly create custom content filters for your organization's unique needs. As AI becomes part of every application, it's also critical that we can govern and monitor it the way we do all software. That's why we've partnered with HiddenLayer, and now open models in the model catalog are scanned with HiddenLayer model scanner, so developers know they are secure from the start.

Microsoft Purview enables you to bring data governance into the AI application to help you control and monitor the use of sensitive data. Finally, threat protection in Microsoft Defender for cloud has native integration with Azure AI Content Safety, so SecOps teams get automatic alerts for jailbreak attacks, sensitive data leaks and other threats to quickly investigate and respond. With Azure AI you have an end-to-end suite of tools to make AI systems that are trustworthy, safe and secure. I can't wait to see what you do with it. Back to you, Eric.

ERIC BOYD: Thanks, Sarah. Let's take this innovation real and look at how one organization, Epic, is putting generative AI to work. As Satya mentioned yesterday, Epic is a leader in healthcare technology whose software is widely used by hospitals, clinics and healthcare systems globally. The breadth of Epic's application of generative AI includes more than 60 use cases. They are making material progress in increasing physician efficiency and the quality of care for patients worldwide. Let's take a look at two scenarios.

(Start video segment.)

EPIC DEMO: Let's see how Epic is helping physicians manage patient messaging more efficiently. Our patient, Joseph, has had a heart valve replacement, and he is wondering if spending time with his granddaughter poses a risk to his immune system. He opens Mychart on his mobile device to send a message to his doctor. Our physician, Doctor Walker, receives Joseph's message in her inbox. Doctor Walker leverages In Basket to draft an AI-generated reply to Joseph, which includes a congratulations for the new addition to his family and relevant details and advice. Doctor Walker can review and edit the message for accuracy, as well as add a personal note before sending back to her patient. In Basket is helping triage patient communications across more than 100 institutions. At Stanford, it has reduced physician burnout by 17%, and at Mayo Clinic, it has reduced time nurses spend on each message by 30 seconds.

Next, let's look at how Epic is helping physicians accelerate discovery. In SlicerDicer, we open a chat in Sidekick to query patient data. Let's start by asking about hospital admissions for kids with VTE. SlicerDicer quickly answers our question and updates the patient population. It is also translated to natural language in our question into medical terminology. For example, it knows that "kids" refers to all patients aged 18 or younger, and VTE refers to a diagnosis of venous thromboembolism. Next, let's click medications to further narrow the data. SlicerDicer splits our patient population into two groups based on use of common classes of drugs for treating VTE, DOACs and heparin. SlicerDicer understands that DOACs is comprised of several different medications and constructs a grouping of the four distinct drugs.

We want to better understand how these medications impact patients with VTE. Let's add a measurement to this population to tell us the average length of stay in hospital for each group. SlicerDicer parses the data for the two groups of patients, illustrating that patients administered DOACS typically spend less time in the hospital.

We can also ask SlicerDicer to suggest additional queries to continue data discovery with this patient population.

SlicerDicer is helping physicians to quickly dissect large volumes of patient data using natural language and AI to surface intelligent insights and accelerate discovery.

(End video segment.)

SCOTT GUTHRIE: Thanks so much, Eric, Seth and Sarah. And now that we've covered Azure AI, let's talk about our data platform.

Now, all the AI innovation that you've seen so far is built on a foundation of data, data really is the fuel that powers AI. And our Microsoft Intelligent Data Platform provides customers with the broadest capabilities spanning databases, analytics, business intelligence, governance and AI. And as you heard yesterday from Satya, we're investing in data innovation across this entire platform.

To share more about our data platform vision, I'd like to invite Arun Ulag, who leads our Azure Data Platform group, to the stage. Please welcome Arun.

(Applause.)

ARUN ULAG: Thank you so much, Scott. This is such an exciting time to be in data. This is such an exciting time to be at AI. I'm really, really happy to be here.

With the Microsoft Intelligent Data Platform, we have a comprehensive portfolio of products across the data and AI stack. By working with Microsoft, you not only get really strong capabilities in each area, but even more importantly, you can lean on us to make everything work together seamlessly for you, so that you can focus on moving your business forward.

We also work closely with partners such as Snowflake, MongoDB, Oracle and Reddis to ensure that their industry leading products are available as managed offerings on Azure. Let's go deeper on databases.

In the era of AI, there are many demands on modern cloud-native databases. Data is both structured and unstructured, traffic is globally distributed and traffic patterns change constantly. Your AI models need to work with your data in your databases in real time. That's why when you look at Microsoft's cloud database in Azure, we provide the most comprehensive portfolio across relational, non-relational, open source and caching solutions.

The growth and momentum of Azure databases is staggering. Here are just a few examples.

Customers such as American Airlines have leveraged Azure SQL to modernize and transform their customer interactions, handling tens of millions of calls per day and helping passengers get to their flights on time. You also saw earlier in the presentation, ChatGPT, the fastest growing consumer product in history, is powered by Cosmos DB.

We have several exciting announcements for you today. I'm delighted to announce three experiences that are now enhanced with Copilot capabilities. First, Copilot in Azure SQL will provide self-help for managing and operating your database efficiently. Second, Copilot will help developers write T-SQL queries for Azure SQL by simply using natural language. And third, we have added chat to answer developers' questions about Azure databases for MySQL by summarizing technical documentation.

Let's go over to Cosmos DB, which is quickly becoming the database of choice for the world's AI workloads.

Today, I'm excited to announce Vector Search for Azure Cosmos DB for a NoSQL API. Vector Search makes it easy to search for information by using the semantic meaning versus exact predicates. You can ask questions, like give me books like this other book that I'm interested in, and Cosmos DB takes care of the rest. It's a key requirement for you to efficiently build RAG models. And because Vector Search is simply built into Cosmos DB, you will experience incredibly low latency and no additional costs.

Moving on to PostgreSQL, there are two announcements to share today. First, to enable developers to plug PostgreSQL data into Azure AI, the Azure AI extension for PostgreSQL is now generally available. This enables Postgres developers to invoke Azure AI services and seamlessly pass data to models such as Azure OpenAI, and the Azure AI language service.

Second, the preview of in-database embeddings for Azure Database for Postgres, this new capability brings Microsoft open source E5 text embedding models directly into PostgreSQL, which not only reduces latency, but also drives very high (inaudible) throughput.

Now, let's move on to analytics.

In our Build conference last year, we announced the public preview of Microsoft Fabric, our new data platform built from the ground up for the era of AI. And just six months ago, last November, we announced that Fabric was now generally available.

Fabric gives you everything you need to go from raw data to AI or BI value in the hands of your business users in a single, integrated SaaS platform. Fabric is truly unified at the most fundamental level, with unified compute and storage, a unified experience, unified governance and a unified business model. And this unification dramatically accelerates time to value.

Importantly, Fabric also unifies the business model across all analytics workloads. For example, overnight, your Fabric capacity might power data engineering and maybe some data science. And in the morning, as people walk into the office, that capacity moves to Power BI and to SQL. And this unified business model helps you significantly reduce your costs.

Fabric's vision has really resonated with customers, and the momentum we have seen since Fabric became generally available just six months ago has been stunning. Fabric today has over 11,000 paying customers. Let me give you a couple of examples.

Dener Motor Sports, who runs Porsche Cup Brasil, uses Fabric to make decisions based on real-time telemetry that's literally streaming in from cars as they race around the track.

One New Zealand, the largest telco in New Zealand, was able to get an end-to-end solution built and deployed into production in just three weeks.

Well, we have more exciting news for you today. I'm delighted to announce Real-Time Intelligence, which is a major upgrade to Fabric.

(Applause.)

Thank you. Customers have tons and tons of real time data streaming in. It could be from IoT devices on factory floors, or cloud application telemetry or security logs, for the matter, just as a few examples.

And real-time data is notoriously hard to work with. I know Fabric will make working with real-time data drop dead simple. And Real-Time Intelligence in Fabric is an end-to-end experience that makes it simple to gather, unlock and act on real-time data. And just like the rest of the Fabric, we will support data from everywhere, of course, the Microsoft Cloud, but also AWS Kinesis, Google Pub Sub and Confluent Kafka.

Let's look at how these capabilities have come together. Let's roll the video.

(Begin video segment.)

DEMOER: As organizations have modernized, they've begun to generate massive amounts of digital exhaust, spanning manufacturing, supply chains, application telemetry and more. These

streams of real-time data are ready to be collected and analyzed and harnessed to drive improved business outcomes.

Microsoft Fabric enables organizations to create end-to-end analytics solutions spanning the breadth of an organization's data. OneLake in Microsoft Fabric allows them to unify their entire data estate, so all of their data can be discovered, governed and easily integrated into data solutions. And now, with the Real-Time hub in Fabric, this includes unifying streaming and real-time data as well.

The new, Real-Time Intelligence capabilities of Fabric make it easy to bring in streams of data from a ton of different sources, from both Azure and cross cloud, from sources like Amazon, Google and Confluent.

In this case, we're going to connect to an Azure IoT hub to bring in a stream of data tracking delivery trucks and package deliveries, so we can incorporate this data into a new, real-time analytics solution we're building in Fabric. With just a couple of clicks, I can add the data stream to the Real-Time hub. Immediately, we see the data flowing in. As we ingest the data, it can be transformed and routed to output destinations and Fabric.

Switching over to one of these output databases, I can easily set up shortcuts to other data in Fabric to enrich it for my solution. I can also quickly write queries to join data together, such as pulling together delivery destination information to combine with our IoT data stream. And, of course, Fabric makes it easy to create visual, interactive, real-time dashboards so everyone can keep track of the data as it updates.

Now, if we switch back to the Real-Time hub, we can see how you can bring together a wide range of data streams, all unified in Fabric. This opens up a ton of new opportunities for downstream usage of the data by more personas through Fabric. For example, if I open up the deliveries data stream, it's easy to reason over the data and set up alerts using the data activator experience in Fabric.

Here, I can see the sensor telemetry data for our package deliveries. With just a single click, I can turn on AI powered alerting. This will look at the values in the stream and automatically spot anomalies in the data.

If I drill into the sensor alerts on the packages, I can see a sample of packages being tracked where a few clearly have some outliers. The AI has automatically detected these anomalies, and if I scroll down, I can see a summary of the total number of anomalies detected across all of the packages being tracked. I can also control the sensitivity I want the AI to trigger alerts on.

And finally, I can easily configure what action I want to be taken each time an anomaly is found. I can set up custom actions to trigger code, such as using an Azure function. And this is fully integrated with Power Automate, so I can drive action into operational systems. In this case, I'll just have it generate a Teams message to see things running end-to-end.

Prior to the real-time capabilities of Fabric, making use of streaming in real-time data required deep expertise and complex tooling. Now, with the Real-Time hub and Fabric, we're democratizing the full end-to-end experience for how people ingest, manage, analyze and act on real-time and streaming data.

(End video segment.)

(Applause.)

ARUN ULAG: Thank you. Since the Fabric launch, we have welcomed software development partners to join the Fabric ecosystem by integrating their products directly into Fabric. For customers, it means that Fabric gets tons and tons of new capabilities from industry leading data and AI companies that are just simply a part of Fabric.

ESRI is the world's leading geospatial company, and they're bringing the entire mapping and geospatial analytics platform to Fabric. Neo4j is a world-class graph database, and they will enable Fabric customers to build gen AI applications that are powered by knowledge graphs.

One partnership that I wanted to highlight is the deep collaboration between Microsoft and the London Stock Exchange Group to bring LSEG's financial market intelligence offerings into Fabric as a first class experience. This allows financial services customers worldwide to get more value from LSEG's high value financial data and get unique data transformation capabilities. Let's see a demo of how LSEG's integration into Fabric will work.

(Begin video segment.)

DEMOER: I'm an analyst working on a project to improve the risk evaluation we use for a company's investment portfolio. There's some data I want to incorporate from LSEG, a leading financial data provider. Fortunately, LSEG is now integrated into Fabric through the Workload Hub.

Here in the Workload Hub, I can see a wide range of software development companies that have brought their capabilities into Fabric, including LSEG. And if I click on the LSEG workload, I get an overview of the capabilities they provide, and I can see this includes their data catalog which will be added to Fabric.

And from here, I can navigate to the Workload home page for LSEG. This is just like Fabric's built-in Workload homepages, but this one is tailored to LSEG. From here, I can access their data catalog, and do LSEG perm ID minting, which is needed to link the LSEG data sets to other data in Fabric.

Let's open the data catalog where, if a user is entitled to access and use LSEG data, they can search for it. Let's search for environmental, social and governance rating data, and here's the data I'm looking for.

Now, with just another click, I can add it to my lake house.

And now, switching back to the lake house, I see the ESG score data has been added. Now that the data is in Fabric, I can use it to create my solution. I have it joined together with the rest of my data. I can work with it in a data science notebook to explore and develop AI models, and I can create interactive Power BI reports to share with my team.

Now, I have the extensive data of LSEG, integrated into Fabric.

(End video segment.)

ARUN ULAG: Today, I'm happy to announce the preview of the Microsoft Fabric Workload Development Kit. This enables any software development company to extend Fabric by bringing in their own experiences, just like LSEG and all these other larger software companies are doing. The Workload Development Kit gives you massive reach into the rapidly growing Fabric, customer base, and the ability to monetize and grow your business through the Azure Marketplace.

With the launch of Microsoft Fabric, we introduced OneLake as a single, unified, SaaS data lake for the entire organization. Data lakes, as we all know, can be messy and complicated, and OneLake makes data lakes really easy to use, not just for developers, but also for business users.

With OneLake, Microsoft has moved away from proprietary data formats to open source data formats at the heart of Fabric. All of the structured data in Fabric is based on Apache, Parquet and Delta Lake, so it's completely open.

In addition, we have taken a couple of years to optimize all of the Fabric engines, Spark, Data Warehouse, Power BI, real time to provide industry leading performance based on Delta Parquet. This lets all analytics engines work seamlessly with a single copy of your data.

We also recognize that data lives everywhere, not just in Azure, but also in AWS, GCP and on premises. And OneLake can work with your data without first having to migrate everything to Azure.

Now, one of our most strategic partnerships, leveraging the strength of OneLake, is with Databricks. Azure Databricks and Microsoft Fabric are highly interoperable as both products are based on an open lake house architecture. Delta Parquet is the underlying data format for both Databricks and Fabric, making both systems highly interoperable.

Last November, at Ignite, Scott and I were joined on stage by Ali Ghodsi, the CEO and co-founder of Databricks, to talk about our partnership. We're really proud of the co-innovation that we have with Databricks, and we have a robust roadmap of innovation that we're bringing to a joint customers.

Today, I'm delighted to announce the general availability of Vector Search in Azure Databricks, which allows you to create a vector search index directly from a delta table to easily build generative AI models natively in Azure Databricks.

I'm also excited to announce that soon, you'll be able to access Azure Databricks unity catalog tables directly in Microsoft Fabric, making it even easier to unify Azure Databricks with Fabric. This data can be read like any other data in OneLake. You can write SQL queries. You can use it with any of the workloads in Fabric, including Power BI. And when data is modified or tables are added, removed or renamed in Azure Databricks, the data in Fabric will always remain in sync.

You can see how Databricks and Fabric are working together to ensure that you have an open and governed data foundation on top of OneLake.

Well, we're not stopping there. Today, I'm pleased to announce our expanded partnership with Snowflake. (Applause.) Thank you.

With this partnership, Snowflake will also be able to natively store their data into OneLake. And since OneLake is already integrated with all Fabric engines, Power BI, but it's also integrated into Azure AI Studio, Microsoft 365, data from Snowflake will still seamlessly flow through the entire Microsoft ecosystem. Our vision of a deeper integration between Fabric and Snowflake will start rolling out later this calendar year.

Now, many of you know that Snowflake has embraced Apache Iceberg, which is another open source data format that's an alternative to Delta Lake. We're excited to announce that we're taking a commitment to open standards further by including support for Apache Iceberg, in addition to Delta Lake and Fabric.

Now coming soon, all tables stored in OneLake will be automatically available in both Iceberg and Delta Lake formats. And we believe that the translations between these formats should be open as well, which is why Microsoft is actively contributing to the open source Apache XTable project.

Now, here's a quick sneak peek at what the Fabric integration with Snowflake will look like. Let's roll the demo.

(Begin video segment.)

DEMOER: My organization uses both Fabric and Snowflake. And now we're starting a new data warehousing project in Snowflake for our customer loyalty program. But we want to keep our data and workstreams unified across both places. Our sales, customer support and order systems are in Fabric, and this new project needs to unify all of this data.

Fortunately, with new integration between Snowflake and Fabric, we can work seamlessly across both systems. Let's start in Snowflake and see how this works.

I'm going to create a new database for some customer loyalty program data, and right off the bat, you see a new option to use Iceberg tables. Iceberg is an open format that we're now also supporting in Fabric.

And along with this, Snowflake will soon allow you to pick Microsoft Fabric as a storage provider. When selected, Snowflake will directly store the data for the database in OneLake in Fabric.

Let's add some data to the new database, and we can see it defaults to storing the data in OneLake in Fabric. And keeping things simple, I'm just going to upload some CSV data, review the column mappings, and now my data is loaded. I can see my loyalty member data here in Snowflake, but now, let's switch over to Fabric.

And here you can see something really cool has happened. Automatically, a new Snowflake item in Fabric has been created. This is a fully integrated item that works just like every other data item in Fabric. When I open it, I can immediately see the data that was written by Snowflake into OneLake.

Now, I'm going to bring in the other data my project needs from Fabric, and I can do it right here in the Snowflake item. I can shortcut to the data I need from Fabric, so there's no copying of data. And if I jump ahead, you can see I've linked orders, sales, customer support and review data into the Snowflake item.

And by doing this, automatically, the metadata for these tables is now being expressed in Iceberg format. This means that when I go back to Snowflake, these tables from OneLake are now available here as well. This lets to my team work fully bidirectional between Snowflake and Fabric, so I can write a query here in Snowflake to join the data together, occurring directly from OneLake, so my Snowflake solution is working on one copy of the data.

Now coming back to Fabric, I can also use all the Fabric workloads with the data. For example, I can create a new notebook and even use Copilot to quickly analyze the data.

Copilot, along with every workload in Fabric, is working on the same one copy of the data. And now that we have all these different data sources joined together, let's ask Copilot to help find outliers between orders, support calls and sales.

I can insert the code Copilot generated into the notebook and run it. And immediately, I have a pretty interesting visualization where I can see some outliers. And, of course, the data can also be used directly in Power BI, and I can even connect to the data from within Excel, since the OneLake data hub is natively built into Excel.

Most importantly, since Fabric and Snowflake are using the same Iceberg format of the data and collaborating through OneLake, my team can seamlessly collaborate from both while always using one copy of the data.

(End video segment.)

(Applause.)

ARUN ULAG: Thank you. To share more about the incredible work that Microsoft and Snowflake are doing together on behalf of our customers, I'd like to welcome Christian Kleinerman, EVP of Product at Snowflake, to come up on stage.

(Applause.)

Welcome, Christian. Thank you so much for joining us. It's great to have you here.

CHRISTIAN KLEINERMAN: Super excited to be here. Hi, everyone. We've been partners with Microsoft for a long time, with Azure being a choice many of our customers make. And we also have many product integrations like Azure AI. You and I have been personal sponsors of our integration in Power BI. We're very happy with the announcement today to continue to deepen our partnership.

ARUN ULAG: Thanks, Christian. Now, Snowflake has clearly been a leader in the data and AI space for quite some time. Can you talk a little bit more about what's going on at Snowflake, and how are you thinking about open formats and our opportunity together?

CHRISTIAN KLEINERMAN: Certainly. At Snowflake, we prioritize innovating to address our customers' most pressing needs. That leads to a focus on governance and security, ease of use and enabling organizations to collaborate on data. And, of course, we're moving fast to unlock the opportunity that AI represents. We do this by extending what our product can do, and by offering AI for development of data applications through our fully managed service, Cortex AI.

Related to today's announcement, a big part of what large enterprises are seeking is interoperability. We saw a lot of demand and organic momentum behind Iceberg, and that's why we chose Apache Iceberg to deliver that interoperability. And we are fully committed to open storage. With the support of Iceberg and Fabric, we can envision mutual customers being able to interoperate between our technologies with less friction.

ARUN ULAG: This is definitely an exciting partnership for both companies, Christian, because this means that all of the Snowflake data in OneLake will be available to the entire Microsoft ecosystem. That's Power BI and Fabric for sure, but also because OneLake is deeply integrated everywhere, the data will be built into Excel. It'll be built into Teams. It's in Azure AI Studio. It's in Copilot Studio. It's basically everywhere our business users are. And this is also true bidirectionally as well, as we saw in the demo, because Snowflake will also be able to leverage the data in OneLake.

Any thoughts in terms of the opportunity that we have going both ways?

CHRISTIAN KLEINERMAN: Certainly. We are doing this to deliver a better experience to our mutual customers. Some organizations have chosen to leverage both Microsoft and Snowflake, and they need open standards for storage to be able to accomplish that, and we are committed to supporting that.

Our joint goal is to help these organizations reduce friction, eliminate the duplication of data and enable them to be wildly successful with whatever mix of technologies they choose. And that could be, for example, processing data with Cortex AI or Snowpark, but also keeping that data open and accessible, as you say, for Excel or maybe activating with Teams.

Iceberg already has a very broad ecosystem, and today it just got dramatically stronger thanks to Microsoft's support. We're so excited about this opportunity, and we look forward to more from this partnership.

ARUN ULAG: Thank you so much, Christian. I think Azure is the fastest growing cloud, right?

CHRISTIAN KLEINERMAN: Fastest growing on Snowflake, so keep up the momentum.

ARUN ULAG: Awesome. Thank you so much for joining us. I appreciate it.

CHRISTIAN KLEINERMAN: Thank you for the partnership.

(Applause.)

ARUN ULAG: Hopefully, you get a sense of how the Microsoft Intelligent Data Platform gives you a comprehensive set of capabilities to power your AI journey. Thank you so much, and back to you, Scott.

(Applause.)

SCOTT GUTHRIE: Thanks so much, Arun.

Now, as I talked about, data is very much the fuel that powers AI. And you can see with all of those announcements how, with Azure and the Microsoft Intelligent Data Platform, we now have the richest data platform that you can use to send your data and integrate your data with the richest foundational models to build amazing application experiences.

We've covered a lot today already in terms of the application stack, the developer tools, the AI platform, the data platform. Let's now dive deep into the infrastructure that's powering all of this.

We've invested heavily over many years to make Azure the place to do cutting edge AI innovation. Azure is the world's AI supercomputer, and it's the infrastructure that's powering not just ChatGPT and all the copilots and all the AI products and services coming from Microsoft, it's also the AI supercomputer that's being used to train the large foundational models. And we've created purpose-built AI infrastructure to deliver reliability and performance at scale. We now have more than 60 Azure regions live around the world, and we're bringing all that AI scale and power to you.

Now, we know also, with all this power comes responsibility. Microsoft is one of the largest buyers of renewable energy now in the world, and we're on track to meet our goal to have all of our data centers powered by 100% renewable energy by next year.

(Applause.)

We're also optimizing and innovating at every layer of the infrastructure stack, and Satya talked about this quite a bit yesterday. And we're taking the learnings from running the largest and most complex AI and compute workloads in the world, and using that feedback loop to continue to make the best infrastructure. And what this means is when you use Azure in the Microsoft Cloud, you get industry leading price, performance and feature set capability. And we're doing this from systems to silicon.

Microsoft provides now the most complete range of AI accelerators. Alongside our own custom Azure Maia series, we partner deeply with Nvidia and AMD to bring you the latest innovations in AI acceleration, such as AMD's new MI300X, available first on Azure, as well as Nvidia's upcoming H200 series.

And based on the workload and AI use case, we can now dynamically choose which AI accelerator we use for all the built in copilots that we deliver, as well as all yours. And the silicon diversity is what allows us to run the most powerful foundation models, giving you the best AI performance and the lowest cost.

Developers and organizations are seizing the cloud and AI opportunity with Microsoft. And in addition to our horizontal capabilities of the Microsoft Cloud and the Copilot Stack, we also now offer powerful industry tailored solutions. And the Microsoft Cloud for Industry Solutions is built on top of Azure, Microsoft 365, Power Platform and Dynamics 365, and they provide a common data model, compliance and AI workflows, tuned for specific industries like healthcare, financial services, manufacturing and retail.

And we're working with software companies around the world to integrate their solutions with the Microsoft Cloud for these industries to deliver high value Copilot and AI experiences. And earlier in the keynote, you heard about the work we're doing with LSEG in financial services, and Epic in healthcare and the great work that they're doing around AI with Microsoft.

Symphony AI is another great company that's creating industry specific, AI-powered solutions to do things to help investigate financial crime. And they estimate that these investigations can now be completed 60% to 70% faster because of their use of Azure AI.

And Schneider Electric is turning to AI to help solve one of the world's biggest challenges, which is sustainability. And whether it's reducing their own carbon emissions or helping their customers optimize their own energy use, AI is at the center of their ambitions.

Now, Microsoft can both help you build your new AI solutions. It can also help you grow your business using our Azure Marketplace. And the scale of our marketplace is growing rapidly, with billions of dollars of revenue now transacted every year through it. And using our marketplace,

you can now co-sell with Microsoft's enterprise sales force and further accelerate your business growth.

Now, safety and security are foundational to everything we do, and we've touched on these topics several times already throughout the keynote. Developers need to deeply integrate security into every part of the development lifecycle.

What I'd like to do now is invite Julia Liuson, who's the president of the Microsoft Developer Division, and John Lambert, security fellow at Microsoft, to the stage to talk about how we're applying the security principles at Microsoft, as well as how we're working to enable developers outside of Microsoft to build apps securely, starting from the very first line of code.

Please welcome Julia and John.

(Applause.)

JULIA LIUSON Well. Thank you, Scott. My name is Julia Liuson. My team built some of the most widely used developer tools in the world, with products like Visual Studio Code, Visual Studio and GitHub. So it's very important to me that we must continue to earn and maintain the trust for all of our customers.

JOHN LAMBERT: And I'm John Lambert, corporate vice president and security fellow on the Microsoft Threat Intelligence team.

JULIA LIUSON: So today, John and I are going to deep dive into Microsoft's Secure Future Initiative that we launched last November and further extended earlier this month. As Satya mentioned in his keynote yesterday, there are six pillars in our security initiative. And today John and I are going to really focus on two pillars which are particularly relevant to developers. It's the protecting identity and secrets and protect your engineering system.

Now, over the years, we have done a lot of work for each one of these pillars, but in response to the evolving threat landscape, we have made security the absolute top priority. We're very committed to prioritizing security above anything else, and for all of Microsoft.

JOHN LAMBERT: While developers chart their career milestones by the products that they ship, but security people, our milestones are incidents. For me, that's SQL Slammer, Stuxnet, SolarWinds. It's just very important that we learn from these incidents. And so SFI has three core principles at the heart of it, secure by design, secure by default and secure operations.

It should be the case that developers don't have to do a ton of work in order to secure what they're doing. We want them to fall into the pit of success, and that starts with these three principles.

Now, Julia, you and I have worked together for like a decade, but I feel like in the last three months, we've spent more time together than in the entire 10 years. I guess there's just no better way to speed the learnings from incidents to improvement than just to work together every day.

JULIA LIUSON: Yeah, and as important as all of the technical work we are going to embrace, security really is a team sport. It's going to take a culture change for everyone at Microsoft and in the industry. Instilling a security-first culture will be critical for us to engage the entire company and industry in this problem space.

We are sharing our learnings today because many of the challenges we face are really not unique to Microsoft. It really affects most of the companies in the industry.

Now, John, you have been in the security business for a very long time, and can you share with us some of the changing trends that you're seeing and also discuss the evolving threat landscape, which really provide context for the entire security initiative, as well as the culture change that we want Microsoft and the industry to drive towards?

JOHN LAMBERT: Yeah. Organizations, they don't face threats. They face threat actors. And boy, they have been busier than ever.

At Microsoft, we track over 300 different threat actors from nation state to cybercrime. For them, hacking is their job and they want to be good at it, but they don't want to have to work harder than they have to. And what's easier than hacking in? Logging in!

And so they really focus on password based attacks. And we've seen over 10 times increase in password based attacks just in the last year. All of this activity creates a kind of anti-economy estimated to be worth \$8 trillion. And if it was a country it would be No. 3 in terms of GDP. So they want to work really hard, but at Microsoft we want to make them really, really bad at their jobs.

JULIA LIUSON: And talking about these threat actors, one of them you and I have been spending a lot of time in the last few months working on, and it's called Midnight Blizzard. Can you tell us more about this one?

JOHN LAMBERT: Yes. So first, in our taxonomy of how we name actors, Blizzard is the family name for threat actors originating from Russia, and we spent a lot of time tracking and learning about this actor.

As an intelligence agency, they target government departments and defense contractors, but the second most prevalent vertical they target is information technology, stuff that developers make because hacking that gives them access to all of the rest of their targets.

Hackers deeply focus on the idea of "hack one to hack many," and they exploit chains of trust and transitive access. I've often said that defenders think in lists and attackers think in graphs, and as long as that's true, attackers are going to win. Midnight Blizzard shows a strong command of thinking in graphs, how systems are connected and how they traverse.

And so how do you traverse that graph? Credentials. And one of the things that we've learned from our own incident is that it's very important to treat non-prod to the same security standards as production itself.

Midnight Blizzard targeted developers. They targeted developers to get credentials of developers, and they got credentials that gave them access to demo and test systems. They are using those credentials they are getting to build an attack graph. They want to know where those credentials lead.

JULIA LIUSON: Yeah, because password is just a string. So developers risk accidentally checking them into their source code or emailing it to a colleague. To help mitigate this risk, we built GitHub Event Security with secret scanning. When we work with our customers to turn on secret scanning in GitHub Event Security, nearly 100% of the customers found that they have some secrets in their source code, and sometimes they actually discover thousands of secrets in the first scan.

Another risk with the scanning tool is that it's not perfect. It may not identify every single secret because it's just text. And some of these secrets don't have very obvious ways to identify them.

JOHN LAMBERT: The other thing with secrets in an incident, we also have to rotate them and that means security people have to page developers and we really don't like doing that. And developers really don't like getting those pages. And so that's why it's really important that we move to managed identities that are system managed. That allows us to also like rotate them automatically and then it saves toil, but critically in response it gives us speed.

JULIA LIUSON: Yeah, and so we're taking this learning and applying it to our "protect secrets" pillar. One of the key actions is that we want to ensure 100% of applications are protected with system managed identities.

Now, where we really want to go in the long run is that we want there be no passwords at all, so developers cannot accidentally leave them in source code or email them to their friends, and they never have to rotate them ever again.

JOHN LAMBERT: Yeah, this is going to be challenging because as a developer, like where you go online to learn about how to finish a scenario or watch a video about it, they all tell you use a password to do that.

JULIA LIUSON: It's absolutely true. And most cloud providers have videos and symbols out there that actually take use of no passwords, so those problems are certainly not unique to Microsoft, it is a prevalent issue across the industry. And we are taking great pride in how we build tools to help developers really be very, very productive.

So we are actively working on improving our tools with secure-by-design and secure-by-default principles, which means we want developers to be able to just follow the guidance set by the tool, and the paved path in the tool will automatically use system manager identities, so developers don't really have to think about it.

Now, we do anticipate that by using system managed identities, you will feel a bit more complex because there's a little bit more code and concept than just using a simple text based password. But this is where security-first culture change really need to happen. We must embrace productivity and simplicity with a security-first mindset.

So now, John, this is also not the first time that you have run into Midnight Blizzard, right?

JOHN LAMBERT: That's right. They were also the threat actor behind the SolarWinds incident. SolarWinds is a company in Texas that was the victim of an intrusion by Midnight Blizzard. They compromised their build process behind their product. And so when the product was built, malware from Midnight Blizzard inserted a backdoor into the product.

Now, that malware, that backdoor was never checked into their source repo. It was inserted during the build process so the product would get built, digitally signed, go through update systems to their customers, and if Midnight Blizzard was interested in one of their customers, they followed up with a secondary backdoor.

This shows us that the SolarWinds actor Midnight Blizzard is interested in supply chain attacks. They want to be able to infiltrate the software supply chain in order to get access to end customers. It also makes detecting these attacks very difficult.

So threat actors, they target code, they target organizations, and they target the engineering systems that make the applications that those organizations use.

JULIA LIUSON: So this SolarWinds example really shows the importance of protecting our engineering system, which is a key pillar in our security initiatives. You just spoke about what can go wrong when an intruder attacks the build-and-release environment.

Another very important dimension of keeping our engineering systems secure is to understand the entire surface area of the engineering system. During our security initiative, we have actually deleted over 700,000 unused demo and sample and test applications. We have also deleted a lot of the repos and build pipelines which are associated with these applications.

JOHN LAMBERT: Yeah, getting rid of unused apps and systems. It's just really important for security because threat actors, they just care what's on the attack surface. They don't care if you consider it legacy or modern or it's something you're managing or not managing, if it's got credentials and if it's alive and it's got access to resources, that's what they want to go after. Sometimes the fastest way to de-risk is to get rid of old systems and applications.

JULIA LIUSON: Yeah, exactly. And as we apply the "secure operations" principle there are two really important lessons to improve the overall operating environment.

The first one is that get rid of anything you don't need from your overall engineering system. The second one is that we really need to treat non-production samples and test apps, just like production app was the same rigor.

This is why the build-and-maintain inventory for 100% of the software asset used to deploy and operate Microsoft product is a key SFI action that we have outlined. We want to make sure all of our applications, no matter if it's a sample or a demo app or production, and all of their app repos and build pipelines have clear lifecycle management and policy governance.

John, I think another very interesting topic to discuss is the attacks on the developer ecosystem, particularly the open-source ecosystem. We have been seeing some very sophisticated social engineering.

JOHN LAMBERT: Yeah, attackers have gone from attacking code customer systems to the developer projects themselves. A recent example is the xdg-utils backdoor. This is an open-source project. It's used by Linux based systems. Xdg-utils is a data compression library and it's used by critical things like SSH. And you could understand why a threat actor would want a backdoor SSH.

So this threat actor worked for years to cleverly insert a backdoor into this project, just waiting for the moment where it would get wide distribution. They knew security tools would run against it, and so they cleverly concealed it.

They were almost there. There was just one glitch; the way they made their backdoor, it gave executables just a little bit of a performance slowdown, not a big one, but just a little bit of one, but they kind of met their nemesis.

Their nemesis was a developer with a memory profiler, and so at the final hour, a Microsoft engineer, Andres Freund, he was working and running some regression tests. And he's like, "Why is this thing slower than it should be? It shouldn't be slower. That didn't change; what's going on here?"

And he just kept digging and digging and digging. And then he discovered the back door. He saw something. He said something. And it resulted in an investigation around the world which prevented that backdoor from getting mainstream distribution.

So even though Microsoft wasn't affected by this backdoor, it's just a great example of security culture in action.

JULIA LIUSON: Yeah, and I really love the story because it shows the behavior of a developer who has a security-first mindset. It's also a great reminder for all of us the importance of securing all of our software supply chain.

With each one of these attacks, inside Microsoft, we're always looking at whether our existing systems of managing OSS is good enough to actually help us to defend against these attacks. We're also building our learnings into GitHub Event Security dependency review, with Dependabot, which will help developers to alert whether repo is using a piece of software with vulnerability. It will actually generate PR to help developers mitigate these issues.

JOHN LAMBERT: Yeah, raising alerts is good, but to get them acted upon quickly, you need the right security culture. And so part of that is education. Someone once said that training is for certainty, education is for uncertainty, and security is just full of uncertainty. So yes, take your security training, but also embrace curiosity. There's just so much more that we all have to learn.

JULIA LIUSON: John, thank you so much for sharing insights on how threat actors actually approach their job. I think that we will all need to embrace the new security-first culture. Security is a team sport and it's job one for everyone at Microsoft.

It's super disturbing to see how many known threat actors are actually out there, and how they're actively hacking the entire industry. Defending against threat actors is not only a Microsoft challenge, but also a challenge for every single company. We will require collaboration of our customers and partners to collectively raise the overall security posture.

As we learn more, we will continuously improve our system and tools. We want to also help our customers improve their application code and engineering process as well. As a global provider of software, infrastructure, security and cloud services, we are committed to doing our part to make a world a safer place.

So thank you for being here, being part of our security journey. I can't wait to see what you build. Thank you and enjoy the rest of Build. Back to you, Scott.

SCOTT GUTHRIE: So thank you. Thanks, Julia and John, for convening this important conversation. So to recap what you saw today, Azure and the Microsoft Copilot stack is empowering every developer and every organization on the planet to innovate using AI. I'm looking forward to building some great AI solutions together with you. We have an incredibly exciting future ahead of us. It's never been a better time to be a developer. Thank you and enjoy the rest of Build.

END