05232023 Build Scott Guthrie Keynote

Build 2023
Scott Guthrie
Thomas Dohmke, Seth Juarez, Sarah Bird
Tuesday, May 23, 2023

SCOTT GUTHRIE: Well, good morning, everyone. It is great to be back here at Build and really looking forward to spending the week with you. You know, as Kevin and Satya have talked about today, you know, AI is going to profoundly change how we work and how every organization operates, and every existing app is going to be reinvented with AI, and we're going to see new apps be built with AI that weren't possible before.

And Kevin just walked us through the copilot stack and explained the different layers involved in building solutions that take deep advantage of large AI models. I'm going to build on his talk and walk you through how we're going to make it easy to develop these AI solutions using Azure, GitHub and Visual Studio. We're going to show a lot of code today in my talk and introduce you along the way to some amazing new capabilities that we're announcing and releasing here at Build.

Let's start by talking about Visual Studio and GitHub. The Visual Studio family of products, which includes both Visual Studio as well as Visual Studio Code, are now the most widely used development tools in the entire world across all languages and platforms.

And GitHub is used by over 100 million developers and is literally the home of open source. GitHub also now provides developers and enterprises with an integrated end-to-end developer platform that can be used to collaborate, automate and secure DevOps solutions. And with GitHub you can build amazing AI solutions, including all the copilot capabilities that Kevin just talked about and use it to transform your businesses.

Now, one of the most exciting new capabilities that we've launched with GitHub is a new service we call GitHub Copilot. GitHub Copilot was the first solution that we built using the new transformational large models developed by OpenAI, and Copilot provides an AI pair programmer that works with all popular programing languages and dramatically accelerates your productivity.

This spring, we announced a new update for GitHub Copilot that we call Copilot X, and Copilot X further enhances Copilot, using the latest GPT-4 models and provides immersive Copilot chat functionality, Copilot for pull requests and automatic unit test generation scenarios.

It's truly amazing. And what I'd like to do is invite Thomas Dohmke, the CEO of GitHub, on stage, to show it off in action. Please welcome Thomas.

(Applause.)

THOMAS DOHMKE: Good to see you. Hello, Build. It’s so great to be here in person. I’m Thomas and I’m a developer, and what we have witnessed this past year with GitHub Copilot has been a step change to developer productivity. Software has eaten the world, now it’s AI’s turn. But the autocomplete program has always been just the starting point. We swing the doors wide open to a new age where natural language can power the coding experience. And that begins with GitHub Copilot Chat, a key feature of GitHub Copilot X. So let’s see some real code.

For this demo scenario here, I got some Python code from a coworker. It’s pretty poorly documented. We’ve all been there, right? With GitHub Copilot, I can simply write code by accepting code suggestions. So here’s a little unit test, and I can just press the tab key a couple of times, and I’ve written a full unit test in seven lines of code with no – you know, learning the code and whatnot.

But what I can’t do with Copilot is to provide additional context. I can’t tell Copilot what I want. Well, until now, because now we have Copilot Chat. It sits right here in Visual Studio Code. It suggests tasks based on natural language, and I can interact with it down here to both learn and build while I’m coding.

So I have another file here. It’s called Unknown PY, and it has these three cryptic-like expressions. I have no idea what those are doing. Let’s see if Copilot can explain them to us. OK, Copilot is rendering my response, and you can see the first expression is validating email addresses. The second one is validating phone numbers and the third one is validating strong passwords. That might make sense, I guess.

So let’s highlight this and make this code a bit more readable. "Make this code more readable." And Copilot has analyzed my code, and you can see here now that my regular expressions get nice names. I get nice comments. It looks much better. But I think we can do even a little bit better to make the next person that takes this code understand it even more.

So let’s say we want to separate, "Separate out the validation functions and add more comments." OK, here’s even better code. My regular expressions still have their good names and I get three new methods, you know, to validate email, phone number and password. I think that’s much better. So I can now scroll up here and take this code over. And now I have a much better documented file that I can pass on to the next coworker.

OK, I’ve got a third file, and this seems to be parsing expenses.

(Applause.)

I got a third file, and this one seems to be parsing expenses. Let’s see if this actually works. Oh, I got an error, and I can’t unpack the values. So let’s see if Copilot – see if it can, "Fix the bugs in my code." (Laughter.) And so there’s – you know, there’s a couple of issues. Seems like the split method is splitting the values as a comma separator, but in my comment here, I had said I want the space separator.

And then it also says, you know, the date-time module is not imported, probably because I only highlighted the method. So if you look at here, the fixes and splits, you know, I fix here and split is here, so I can just go across here into the pilot and fix this one method, and now my code runs.

Cool.

So while we are here, you know, I can also ask Copilot to generate some unit tests for me, and I can do this with what we call slash commands. Instead of typing, you know, all this stuff on stage, I can just type "tests," and now Copilot analyzes all the code paths in my code. It looks at the comment, and you can see it here generating multiple unit tests for my class.

In fact, you know, there is one for invalid dates, one for invalid values, command lines, single lines, empty string. It does all that boilerplate code for me, all that boilerplate work for me that I don’t want to do. And now I can take this, click these little three dots here, and insert it into a new file, and have a new unit test file, and it can of course now use Copilot and Copilot Chat to build this out even more.

Well, thank you Copilot for doing all the work for me.

(Applause.)

Now, I know some of you may be thinking, "When can I get my hands on all of this?" And we want to bring this whole experience, all of Copilot X as broadly as we can, and as soon as possible, but right now, there is quite the wait list.

Well, don’t worry, I have an Oprah moment for you. You get a free Copilot Chat, you get a free Copilot Chat, everybody here, right at Build, or in the audience gets early access to Copilot Chat today.

(Applause.)

Use Copilot and Copilot Chat both in Visual Studio and Visual Studio Code. And thank you so much for being here today. Back to you, Scott.

(Applause.)

SCOTT GUTHRIE: Thanks, Thomas. We’re really excited to see what everyone here is able to do with GitHub Copilot. You know, as you just saw, you know, Copilot dramatically accelerates developer productivity. In fact, 46% of all lines of code written by developers that are using GitHub Copilot are now fully AI generated. Seventy-five percent of developers using Copilot are feeling that they can now focus on more satisfying work. You know, no one really loves debugging regular expressions, and you can enjoy your jobs more.

And 96% of developers report being able to complete repetitive tasks a lot faster. And we now have over a million developers using GitHub Copilot around the world today, and again, we’re excited to have everyone here in the live audience be able to take advantage of the latest GitHub Copilot Chat capability, starting this week.

Now we've seen with Visual Studio – now, we've seen GitHub Copilot and Visual Studio in action. Let's now look at how we can use these incredibly powerful tools to start building AI plugins. And as you heard from Satya and Kevin, we're embracing an open plugin standard that provides plugin interoperability across ChatGPT and all of the Microsoft Copilot offerings.

And with GitHub and Visual Studio, we have an incredibly streamlined developer experience for building these plugins. Let's watch a demo that shows all this off.

(Video segment.)

VOICEOVER: Plugins enable developers to integrate custom applications and data into the chat experience for users. Let's say we have an e-retail store that sells outdoor equipment. When a user asks a question about outdoor products within ChatGPT, we might want to provide them with a list of related products that are available.

Now, let me show you how easy it is to create a plugin using Visual Studio Code and GitHub. I could fork and clone the repo on my local machine, but instead I'm going to use GitHub Codespaces. Codespaces are GitHub's one-click developer environment in the cloud. They're fully configured to your repo and they're free for up to 60 hours a month, which makes it a great fit for open-source contributions, exploring new stacks, or working across multiple devices.

Now, let's crack it open and see what's inside. The code for this sample plugin actually looks pretty simple. It uses FastAPI, a Python web framework to expose some rest API endpoints that are called by ChatGPT. For this demo, our plugin is just going to return product information when users ask about them in the chat.

I'll start by just loading some fake products from the products .JSON file. So I'm pretty new to Python and I just love how Copilot helps me as a pair programmer. I'm able to stay in the zone and just focus on the big picture.

Once we're done with our API, all we need to do to transform this project into a plugin is add a little bit of metadata. Well-known AI plugin. JSON has a description of what the plugin does and includes things like a logo that will be shown in the response. The same file can be used to create plugins for ChatGPT, Microsoft 365 Copilot and Bing Chat.

Now, let's add a breakpoint to our API. Once the plugin gets called, I can observe the flow from the request and the response. One of the great benefits of Codespaces is that I don't have to do anything special to install the dependencies. I can run and debug the project, and everything just works.

When our project starts, we can see that Codespaces has opened the ports for our FastAPI server. This is really helpful for debugging web apps and APIs. By default, the endpoints for our codespace are only accessible to me. They’re secure by default, but I can also configure them to be accessible by my team or my organization, or I can choose to make them public.

For the purposes of this demo, I’m going to make our FastAPI port public so that ChatGPT can call it. Once I have the URL copied, I’m going to create a new chat in ChatGPT and then install the plugin via the plugin store. This is really simple. All we specify is the URL to our codespace and ChatGPT will take care of the rest.

Now that our plugin is installed, we can start experimenting with the code inside the chat. To check it all out, all I need to do is start asking questions that might match the natural language instructions we gave in the metadata we exposed for our plugin.

In this case it was climbing products. Because our plugin is running inside of Codespace. We can even interactively debug it as well. As you can see, our breakpoint is being hit inside VS Code as soon as our plugin is invoked. We can step through the code, inspect variables, and really determine if there’s any issues with the way that our code is behaving based on what’s being sent by ChatGPT.

The experience for building a ChatGPT plugin inside VS Code and GitHub Codespaces is incredibly productive. It’s a lot like building a web app. The experience we showed here was for building a ChatGPT plugin, but the experience for building a plugin from Microsoft 365 Copilot and Bing Chat is exactly the same.

Now, let me show you the same plugin we just built running inside Bing Chat. Microsoft 365 Copilot also supports the same plugin, and you’ll hear more about that tomorrow.

(End video segment.)

(Applause.)

SCOTT GUTHRIE: Yeah, plugins are an incredibly powerful way to connect your apps and data to AI. And you just saw how we can use GitHub Codespaces to really deliver an amazing developer environment and Copilot to help you code and actually build it. And as Satya announced this morning, we now have one plugin model extensibility across, again, OpenAI’s ChatGPT service as well as all of our Microsoft Copilot experiences.

And tomorrow morning you’ll hear from Rajesh and Panos who will share more about Teams and Windows and how you can use the exact same extensibility model to build great experiences within them as well.

And you can use this extensibility model to enable plugins within copilot experiences you build within your own apps as well. And a little bit later on in this keynote, I’m going to talk about how you build your own copilot experiences using our Azure set of services.

Now, in that last demo we used GitHub Codespaces to both code the plugin, but also to kind of do a simple test of it within ChatGPT. And because Codespaces is running in the cloud, it was really easy for ChatGPT to connect to it, and you have a nice interactive debugging experience as part of it.

This works really well for getting started and for simple development scenarios, but once you go into production and as your AI plugin gets a little bit more sophisticated, you’re going to want to be able to have a backend that’s going to be able to scale it out and you’re going to need an elastic cloud.

And Azure provides a great set of cloud native services, including powerful Kubernetes as well as container-based solutions that you can leverage to do this. And with our GitHub Actions capability inside GitHub, you can easily set up now a secure cloud-based CICD solution to automate and secure the building, the testing and the deployment of your plugins to these Azure services.

We have lots of organizations out there taking advantage of this capability today. You know, I think a great example given we’re talking about AI scenarios though, is ChatGPT itself, and ChatGPT is a great example of using this GitHub and Azure combination. You know, OpenAI uses GitHub for their development, and they use Kubernetes on Azure, as well as our Azure Cosmos DB database service to run and scale the actual ChatGPT service as well.

And obviously ChatGPT has captured the world’s imagination these past six months. It’s the fastest-growing consumer product in the history of the web. And what to do now is show another demo on how you can also deploy AI apps using that same GitHub and Azure combination as well.

(Video segment.)

VOICEOVER: Now that we’re ready to publish the plugin, we need to run it somewhere that’s secure and scalable for our customers. At the end of the day, an OpenAI plugin is just a web service, so we’re going to use Azure Container Apps to host the production version of our plugin and GitHub Actions to test and deploy it.

Now, I could go through these setup steps in the GitHub UI, but today I want to show you how to configure GitHub Actions in Azure all within the terminal. We’ll use the new Azure developer CLI or AVD for these steps.

AVD is a command line tool that accelerates the process of building apps on Azure by providing developer-friendly commands that help automate each stage of the developer workflow. I’m going to start by making sure I’m logged into Azure.

Next, I’m going to initialize the repo with an AVD template. Templates include everything I need to provision the resources for our plugin and deploy it to Azure Container Apps using GitHub Actions. The component that was added to the repo is the GitHub action.

Now, I always have a hard time reading YAML, so let’s use Copilot to help us make sense of it. OK, it looks like our action is going to provision Azure Container Apps and all the other services we need for our production scenarios. And I also see that it’s referencing some external variables, so I won’t accidentally be checking in any secrets into my repo.

Now, I’m just going to push my changes to the repo. Let’s take a look at what happens when one of these actions runs. I’m going to open the actions log for our repo to see our action running. I can see it’s provisioning all the resources we need and then I can check out the results to get the new URL of our Azure Container Apps where our plugin will run.

Now that we have our plugin running on Azure Container Apps, we’re able to scale to millions of users with the confidence that our plugin will run securely and with high availability. Our continuous delivery pipeline in GitHub Actions enables the team to collaborate and commit changes while ensuring that the updates are tested and deployed in a repeatable, secure manner.

(End video segment.)

SCOTT GUTHRIE: So we walked through, you know, how you can clone a sample AI plugin reop, you know, add code to it, set up a CICD pipeline and deploy it into Azure. And as you’ve seen, everything is pretty easy to do. And what’s even better is you can start doing it today. The exact same repo that we’ve done in all these demos is now public on GitHub and all the capabilities that we’ve shown in terms of Visual Studio and GitHub and Azure are now available for all of you to start building great AI plugins with it. And the beauty is it’ll work with not just ChatGPT but with every Microsoft Copilot experience as well as the copilot experience that you build yourself.

So we spent some time looking at how to develop AI plugins. Let’s now switch gears and talk about a deep dive into what we call AI orchestration.

Now, as Kevin explained earlier, you know, AI orchestration involves grounding, prompt design and engineering, evaluation and AI safety. These are kind of the core fundamentals for how you create great copilot experiences. And this is exactly how Microsoft created all of our copilots as well.

And you know, if you think about the pace of AI innovation that you’ve seen from Microsoft over the last five months, it’s been pretty intense. You know, we’ve introduced GitHub Copilot, Dynamics 365 Copilot, Power Platform Copilot, Microsoft 365 Copilot, Security Copilot, Nuance, Bing and more.

That’s a lot of copilots, and you know, people are asking, "How … what’s in the water? How are you producing so many copilots so fast?" And the reason we’ve been able to move at this fast pace is because we built all of these copilots on top of one platform, which is Azure AI. And this is the exact same platform that all of you can also leverage as you build your own copilot and AI experiences as well.

Azure AI includes several categories of AI capabilities, including our advanced Azure AI infrastructure, and Azure OpenAI Service is our newest capability, and became generally available in January of this year. The Azure OpenAI Service enables developers and organizations to access and use the most advanced AI models in the world, including both ChatGPT and GPT-4, and all of these Microsoft Copilot experiences that you're seeing here at Build are built using these Azure OpenAI AI models. And again, all of these Microsoft Copilot offerings are built using the exact same Azure OpenAI Service and APIs that you can all use to build and scale your own AI solutions as well.

Now, what makes the Azure OpenAI service so powerful is that it enables you to ground and finetune the AI models with your data. We're making it easy to do this with all of our Azure data services, and this enables us to get much more intelligent and tailored outputs from the AI models. And obviously if you do it, you know, one of the questions we get asked a lot is, you know, "Can I … you know, what happens with my data?" And you know, the bottom line is that it's really important as you're grounding AI with your data that you trust your cloud provider.

And obviously Microsoft and Azure AI run on trust. You know, we're committed to doing all the right security work, the privacy and compliance work across everything that we do. And we've been doing this with our cloud for over a decade, and our approach to AI is no different.

When you use the Azure OpenAI Service, your instance is isolated from every other customer. Your data is not used to train or enrich the foundation AI model that's used by others. So you don't have to worry about anyone other than your organization benefiting from AI that's trained on your data.

And your data and AI models are protected at every step by the most comprehensive enterprise compliance and security controls in the industry. And that's why more than 4,500 customers are already trusting and using the Azure OpenAI Service today. It's the fastest growing service in Azure's history. And what I want to do is show a quick video of a couple of customers that have already built amazing solutions with it.

Let's watch a video.

(Video segment.)

VOICEOVER: With Azure OpenAI Service developers can unlock innovative new ways to delight customers and offer completely new experiences with generative AI. As a leading agreement technology provider, DocuSign set out to help customers and signers better understand their agreements.

With a single click, DocuSign extracts key datapoints, creates an AI generated summary of a contract, and even allows users to ask questions and get answers quickly, so people can review and sign agreements faster with more confidence.

Thread helps IT technicians improve customer communication by eliminating tickets in favor of conversations that begin in the applications they use every day, like Microsoft Teams. With Azure OpenAI Service, within two weeks, Thread developed a tool that automates time entries and streamlines resourcing and service request assignments, saving technicians over an hour per day.

Typeface unlocks the creativity of their customers to tell unique stories faster and easier than ever before. Using Microsoft Azure and Azure OpenAI Service, Typeface provides customers like Vasanti Cosmetics with a seamless experience to craft captivating content across marketing channels while staying true to their distinct brand voice and visual identity.

After learning about the Vasanti's brand using images, sample messaging and branding guidelines, Typeface uses Azure OpenAI Service and other services to provide stunning images and compelling copy in seconds, that is right on brand.

Azure OpenAI Service makes the impossible possible for small businesses and startups, as well as large and established brands.

(End video segment.)

(Applause.)

SCOTT GUTHRIE: So you just saw some great examples of companies that are already using the Azure OpenAI Service within their products. As you heard, in some cases it was done in weeks. Let's now dive deep – dive deep into how you can as well.

This week at Build, we're introducing our new Azure Studio. The Azure AI Studio makes it incredibly easy to ground your AI models with your own data and to build retrieval augmented generation or RAG-based solutions. And this enables you to build your own copilot experiences that are specific to your apps and organizations. And you're going to see we've made it really easy to do so.

So please welcome Seth Juarez on stage to actually show us how to do it live. Here's Seth.

(Applause.)

SETH JUAREZ: How's it going, my friends? You're probably sitting here thinking, this has got to be really hard for me to do. I want to build my own copilot. It turns out that creating your own private company copilot that understands your data with Azure OpenAI Service has never been simpler.

In this scenario, we're the Contoso outdoor company, we're trying to make Contoso happen again. Notice that when I ask it a question that's specific about a return policy, because here, let me type it. I'm a fast typer. But notice I'm asking about camping, and I'm wondering if I'm going to get dirty. And now that I purchased the Trail Master X410, I want to know, can I send it back, notice that the response that it gives is effective and it has the right documentation.

How do you do such a thing, you might ask? Well, it all starts with Azure. You go to the Azure portal and create a new Azure OpenAI resource. And this is just like any other Azure resource. What you do when you get this is you get this Azure AI Studio, which is a new experience which makes it incredibly easy to ground your models with your data. I can use any of the available models. Notice here I have GPT-4 and GPT-3.5 Turbo, amongst others that are on there.

Now, this experience allows you also to test it out. I'm using the vanilla ChatGPT here, and notice that when I asked it about camping, it has a pretty good answer. And you can see these responses in there.

The question then becomes what happens when I ask it a specific question about the return policy for the Trail Master X410. Well, notice it just doesn't know.

How do I fix this? How do I put data in there? Well, you've heard a lot from Kevin about something called the RAG pattern, which is where you retrieve facts, inject them into the prompt or augment the prompt, and then what you do is you generate a new response.

Now, the thing that I want to say about this studio, as you're looking at it, is this is your very own Azure AI Studio. It only benefits you. No one else can see your data, this is just for you. This RAG pattern that Kevin talked about is such a common pattern that we've simplified it to just a few clicks. Let me go ahead and click "add data" and then go to "add data source." Look, I got excited and started doing it backstage.

Notice that when you add your data source, you can select from a number of different data sources, all the modalities available to you on Azure. Also with Cognitive Search, as you heard about before, there is a new vector embedding available to you right in there. Also, the cool thing about Cognitive Search is it can crack open your PDF documents and your Word files and be in there.

All right, let's keep going. I acknowledge that I am connecting to an Azure Service. Now let's map some of the data together. There's the content, there's the file name, there's the title. This is the best part of programing, hit "next."

Oh, here's a cool feature. This is a cool feature that we also added into this experience, where you can limit responses to the data content that you retrieve, or you can let the service also augment some of that information as well. I'm going to leave it off.

OK, let's hit "save data," and now I'm going to ask it some of the same questions from before. I'm really concerned about cleanliness and camping, apparently. Who wrote this?

Notice that I'm getting a better answer that's grounded in data. And also, when I ask it about a specific policy, return policy for the Trail Master X410, I want you to notice something about the answer, which is really cool. Notice not only do you get an answer, but you get the citations from your actual data. What happened is when the query went in, it retrieved your data, injected into the prompt and provided the answers.

As you can see, this is pretty simple. Once you've got this working in the playground and your interaction flow is set up, everything is exposed as an API. You can use it in any application and surface it natively in whatever existing application you have, like this one, taking full advantage of the power of Azure AI. It’s pretty cool, right? (Applause.)

SCOTT GUTHRIE: Great. Thanks, Seth. And as you saw, Seth showed you how incredibly easy it is to ground the Azure OpenAI Service, using your own data, keep it private and specific just to your organization so only you benefit from it. And then this enables you to build great copilot experiences that are specific to your apps, specific to your organizations. And the great thing is what Seth just demoed is now in preview for you all to use. (Applause.)

Data grounding is a critical part of how we build applications. As Kevin covered in his talk, AI orchestration also includes prompt engineering. And prompt engineering involves constructing the prompt and the meta prompt that you provide the AI model to produce a stronger, more specific response for the user.

And this week at Build, we’re introducing our new Prompt Flow support in Azure AI. Prompt Flow provides end-to-end AI development tooling that supports prompt construction, orchestration, testing, evaluation and deployment. And it makes it incredibly easy to leverage open source tools and frameworks, like Semantic Kernel and LangChain, to build your AI solution as well.

And so, let’s go ahead and take a look at Prompt Flow in action. And Seth, back to you.

SETH JUAREZ: Let’s do it. All right. Before, remember, we were looking at just the documents. This time, we want to look at a little bit more.

This is the same e-commerce application from before, which could be any SaaS company or product. Notice I’m going to ask it about jackets, but this time I’m going to be logged in as Jane, who has a very specific shopping cart. Remember before, it was just a bunch of documents that were used to augment the prompt. But can we do more? Can we add more information into the prompt?

And it looks like we certainly can. Notice that we’re getting a good response to our data, as well as it’s telling us how well it pairs with the stuff in our actual cart. And they are actually in there, which is pretty cool.

To do this, we are using a brand new Azure feature called Prompt Flow. It’s pretty cool. This is really cool. You should clap a little bit. It’s just so cool. (Applause/laughter.) This is really cool stuff.

This is a powerful suite of tools and services designed for prompt engineering, regardless of your experience in machine learning or AI. It’s the first end-to-end prompt engineering tool of its kind. It includes tooling for prompt construction, evaluation, deployment, testing and even monitoring. With Azure AI Prompt Flow, we can fetch data from multiple sources, and fetch it and put it directly into the prompt, whether it’s structured, unstructured. It’s pretty cool.

I’m going to show you three things. There’s a ton of things in here, but I’m just going to show you three. First, the graph, the prompt and then the model.

This here is the Prompt Flow Graph. And what’s cool about this is you can get structured information coming into here, like a JSON file, and you can flow information down into the prompt. Data flows down into the prompt, like flow prompt, but we called it Prompt Flow. (Laughter.) That’s the first thing.

Let me show you the retrieve cart node, which is pretty cool. I’m going to just show you one node. Notice that this node takes the cart ID and calls this minimal API, and this minimal API is written in .NET. It looks like I closed it. Why would I ever do that? I’m so sorry. It’s a minimal API that you can call that has C Sharp in the backend.

Now, the reason why I want to show you this really quickly is to show you how extensible this is. If I can call an API, I can do anything as I’m doing things in the prompt. In fact, that API that I’m calling that you see right here is exactly the same API that manages the cart on the website. It’s exactly the same thing. I’m not even kidding, it’s the same thing. I could show you the back end, but I forgot to open Visual Studio Code.

The thing about this extensibility that’s actually really cool is that because of that, we can extend to things like LangChain, Semantic Kernel or other tools that are data based, forgive the pun, that can pull data into your prompt, or really anything else.

OK, that’s the first thing I wanted to show you. The second thing I wanted to show you was the actual prompt.

This is the prompt that we’re building. Notice that this prompt is a bunch of text that gets data from the previous nodes. As I zoom in, notice that it’s enumerating the items of the cart and putting it into the prompt. That’s how it knows when it generates the actual response. This then gets put into the model that then, the answer is retrieved. This model can be any of the OpenAI models that you saw before, including any OSS models that you deploy in Azure ML.

All right, so let’s go to the actual model call. This is where we call the model. We can run it directly in here. We can also bulk test and deploy. And if you want to see the responses, you can see the responses right here, which is the answer we saw before, as well as the actual trace of where and how long things take. Finally, you can deploy this, and you can use it in your app as a single endpoint.

In summary, Prompt Flow empowers developers to create richer language experiences that integrate data from multiple sources, providing personalized and contextually relevant information to your users with its focus on simplicity, flexibility and user engagement. Prompt Flow is the first prompt engineering tool of its kind, first end-to-end prompt engineering tool of its kind. I’m excited to see what you build with it.

Back to you, Scott. (Applause.)

SCOTT GUTHRIE: Thanks, Seth. Prompt Flow is an end-to-end system for building, testing and deploying modern, AI-driven applications. And as you just saw, Prompt Flow works not just with our Azure OpenAI Service, but also with thousands of open source AI models as well. And this week Build, we're also releasing a new Azure AI model catalog that makes it incredibly easy for you to also use and consume every open source model out there across your Azure AI Solutions.

Now, Prompt Flow and Azure AI can also help you test and evaluate your applications to help ensure that your prompts are running safely. And in this new era of AI, considering safety is not an optional feature, it's really a requirement. And it can't be an afterthought. You really need to design with safety from the very beginning.

And every AI system that we build at Microsoft is designed to uphold our AI principles. And this week, we're announcing our new Azure AI Content Safety Service to make it easier for you to also test and evaluate your AI deployments for safety, as well.

Now, the Azure Content Safety Service provides the same technologies that we use internally to build our own Microsoft Copilot experiences. And so, you can also benefit from all the learnings that we've had, in terms of making sure that our products are secure and safe.

And so, to share more, let me introduce Sarah Bird, who leads our Responsible AI effort at Microsoft to talk about how we're doing this. Please welcome Sarah. (Applause.)

SARAH BIRD: Thanks, Scott. The recent breakthroughs in foundation models have also transformed what's possible in Responsible AI, enabling us to develop new state-of-the-art tools and technologies that previously, we only could have dreamed of. We've built on top of these AI models deep understanding of text, image and multi-modal content, and have developed a new Azure AI Service to automatically detect undesirable content, making it easier to keep online communities, applications and AI systems safe.

We know from our own experiences building GitHub Copilot, Bing Chat and Microsoft 365 Copilot that safety is a key part of building any copilot. We use a defense in depth or layered approach to safety. First, we've worked with OpenAI to build safety directly into the model so it can recognize problematic queries and respond appropriately.

Now let's talk about the next two layers, the layers that you control, the safety system and the meta prompt. I'll start by showing you the safety system, Azure AI Content Safety. Let's take a look.

(Video segment.)

SARAH BIRD: Azure AI Content Safety is an API-based product that can be deployed as part of any application to monitor for harmful content in real time. I'm in the studio so that we can explore how it works interactively.

For the Contoso bot, most people are going to be asking appropriate queries about products they'd like to buy. For example, here's a user query looking for an ax to cut a path in the forest. If we ask Content Safety to score this, we can see that it recognizes it as safe.

However, one of the challenges with language is that small changes can completely change the meaning. But the power of new foundation models and Azure Content Safety is that it can understand these differences. If I change even a single word in the sentence and rerun the query, you can see that Azure Content Safety understands that this query is not safe, and it would reject it.

Here, I'm using the default settings, but if I want, I can adjust the severity levels to meet the needs of my application and deploy this through the API in production. Contoso uses Azure OpenAI, so Azure AI Content Safety is already built in. Returning to the Contoso bot, I can test these same queries again.

Here, I'm going to start with a recommendation for cutting a path. As you can see, the Contoso bot is prepared to recommend products the customer can buy. However, if I pretend to be a user with the same harmful query as before, I can see the safety systems in action and my bot knows to reject it, protecting Contoso and its users.

(End video segment.)

SARAH BIRD: Pretty cool, right? (Applause.) If you're using an open source LLM, you can call Azure Content Safety directly using the API. However, if you're using Azure OpenAI, you're starting with the first two layers of safety from the beginning. Regardless of what model you're using, the next area that you want to focus on is the meta prompt, that system level prompt that you feed into the model to control its output.

Here we have an example of the safety portion of the meta prompt for the Contoso application that Seth showed. As you can see, it's simply instructing the model in natural language to control for grounding the response, for controlling the tone, for safety, and to prevent attempts to break the model safety systems.

But what I really want to show you is how you can engineer and experiment with these kind of meta prompts, using Azure AI Prompt Flow. Let's take a look.

(Video segment.)

SARAH BIRD: I'm going back to the Contoso retail flow that Seth showed earlier to experiment with the meta prompt and add the safety section I just showed you. Rather than just push it to production, I also want to test it first. Built into Azure AI is a system for testing prompt variants.

As you can see, I'm testing two variants here. Variant Zero contains the new safety section. Now I'm going to use Prompt Flow to see how these two compare. Prompt Flow provides built-in metrics, which makes it easy for me to evaluate. Let's look at the results.

First, I’m going to look at the outputs of the system to understand how these two prompts stack up qualitatively. I can see immediately that there are differences in how they behave. And if I go to example three, which is the example of a user trying to jailbreak the system, I can see that Variant Zero outperforms Variant One because Variant Zero refuses to give information about its prompt, while Variant One is successfully jailbroken.

Now that I’ve looked at the examples, I want to understand how these two compare overall by looking at metrics. I can see that Variant Zero is outperforming Variant One across the board with higher scores in grounded-ness, relevance and coherence, which means it’s the clear winner for me. And now, I can confidently deploy to production and bring the power of AI to Contoso shopping experiences.

(End video segment.)

SARAH BIRD: Here, you’ve seen the combined power of Azure AI Content Safety integrated in Azure OpenAI, and meta prompt evaluation in Prompt Flow to build safety into Copilot. This is just a glimpse of some of the new Responsible AI capabilities we have in Azure AI. We’re making it easier to build real-world, mission critical, safe and secure AI solutions that you can trust and use in your business.

Scott, back to you.

SCOTT GUTHRIE: Thanks, Sarah. Thanks so much, Sarah. In this new era of AI, as I said earlier, safety is not an optional feature. It is a requirement. It can’t be an afterthought; you need a design with safety in mind from the very beginning. And with services like the Azure AI Content Safety Service, we’re providing you with powerful capabilities that are built into Azure AI, automatically integrated into Azure AI already for you, that’ll help you be safe and secure as you build your AI solutions.

Now that we’ve covered the Azure AI and the copilot stack, let’s now dive deep on to our Azure AI infrastructure.

We’ve invested heavily over many years now to make Azure the place to do cutting edge AI innovation. Azure is very much now the world’s AI supercomputer, and it’s the infrastructure that’s powering not just the ChatGPT experience, and all of the AI products and services coming from Microsoft, but it’s also the AI supercomputer that was used to train the large foundation models that OpenAI produced.

And to do this, we built purpose-built AI infrastructure to deliver reliability and performance at scale. We built the largest AI model training system in the world, one with tens of thousands of interconnected GPUs and fast networking.

The picture here is one of our new NVIDIA Hopper-based GPU systems. Azure was the first cloud provider in the world to deploy and offer Hopper-based GPU systems to customers.

In the picture here, you can see a lot of cables. These are InfiniBand cables and they're really important for our GPU clusters. And it's one of the ways that we really differentiate ourselves from others. Neither AWS nor Google support InfiniBand today. And InfiniBand gives Azure the highest network bandwidth and lowest network latency of any cloud hyperscaler in the world today.

That picture shows you what the servers are. And let's switch gears now and take a look at what one of these datacenter facilities looks like.

The picture in the top right, you can see one of our new datacenters that we're building to add to our Azure Cloud region in Dublin. And those little things are tractors and cars, so it gives you a sense of the scale. And that building is just one datacenter that we're commissioning and adding into that region.

And to give you a sense of the scale that Azure is now operating at, we'll now bring live more than 120 datacenters this calendar year alone. That's one new datacenter every three days. And most of these datacenters, like this one, are ones that we're designing and building ourselves. And multiple datacenters then make up what we call an Azure region that you can deploy across.

Now, our Azure region in Dublin is made up of many, many datacenters like this one, and in total size, is already over a mile and a half long and half a mile wide, and it already has 2 million square feet of datacenter floor space that's already live, that you can deploy your workloads in today. And to put 2 million square feet in perspective, that's the equivalent of 35 football stadiums in size, all just in one region inside Dublin.

And Dublin is less than 3% of our total datacenter footprint around the world today. We now have more than 60 Azure regions live. Dublin is just one of those dots there on the map. It's more regions and more countries than any other cloud provider. And we're bringing all of the scale and all this power to you so that you can build amazing solutions on top of it.

But we also know that with all that power comes responsibility. We started our sustainability journey at Microsoft more than a decade ago, and we're committed to leveraging 100% renewable energy by 2025. And our datacenters, including all of the new AI infrastructure that we're deploying, are very much part of that commitment.

As an example, our datacenters in Sweden use free air cooling to reduce carbon emissions, while in Arizona, we're using solar power. In Finland, we're reusing heat energy produced by the datacenters to warm homes in the winter in Helsinki, to help further reduce the carbon emissions from the local community, as well.

And Microsoft, including all of our Microsoft Cloud datacenters, will be carbon negative by 2030. And by 2050, we'll have removed all historical carbon emitted by Microsoft since our founding in 1975. (Applause.)

Now, the Azure AI infrastructure is being used for many, many different AI models and scenarios. We're talking at this conference and today a lot about the great work we're doing with

OpenAI and LLMs on Azure, but we’re also using this exact same infrastructure for many other AI models and experiences as well. And another unique AI partnership we have is with NVIDIA.

NVIDIA is bringing its AI software to Azure. By integrating NVIDIA’s AI Enterprise with Azure AI, organizations are going to be able to have a secure, enterprise-ready platform to quickly build and deploy custom machine learning models. And NVIDIA’s Omniverse Cloud provides a full stack environment to design, develop, deploy and manage industrial metaverse applications. And this gives enterprises access to the full stack suite of Omniverse software applications and NVIDIA OBX infrastructure to scale and combine it with the scale and the security of Azure. Azure is the only cloud provider offering Omniverse cloud today,

And with Omniverse Cloud in Azure, organizations like BMW are now able to build truly virtual factories. The BMW team can aggregate data into massive high performance models, connect their domain-specific software tools, and enable multi-user live collaboration across factory locations. They can run real-time digital twin simulations, virtually optimizing layouts, robotics and logistics systems even years before a factory opens. And it’s all possible from any location on any device.

Now, all the AI innovation that we’ve seen today is built on top of a foundation of data. Data is really the fuel that powers AI, and AI is only as good as your data. And so, it’s never been more important to have a great data analytics and data management foundation in place. And this week at Build, we’re excited to announce Microsoft Fabric, a unified platform for data analytics, really designed for the era of AI.

With Microsoft Fabric, we’re bringing together Power BI, Data Factory and the next generation of Synapse. And Fabric unifies all these analytics tools at every layer into one seamless product with one experience in one common architecture. You literally have unification at every layer, unified data, a unified product experience for all your data professionals to collaborate in, unified governance and security, providing a single source of truth for everyone in the organization and a single way to secure it, and a unified business model to help ensure that all your resources are used in the most cost-effective way.

And this complete platform really delivers data analytics as a Software as a Service model, just like Microsoft 365, making it much easier to stand up and manage with frictionless onboarding, automatic optimization and integration, built-in security governance and compliance. It’s lake centric and has an open data architecture, and it has deep integration with Microsoft 365.

And what’s great is it’s designed, in particular, for the era of AI with built-in copilot support. And this copilot support enables you to use AI to understand and reason over data in extraordinary new ways, while I do show a video of some of these new capabilities in action.

(Video segment.)

VOICEOVER: Let’s take a look at how Copilot and Power BI is empowering everyone to quickly find and share insights.

For example, say I'm an HR analyst who is constantly handling requests from my management to new analysis and provide answers. It can take hours and even days to respond with new reports.

Now with Copilot in Power BI, I can simply describe the report I want to build and get insights in seconds. Maybe we need a breakdown of some of our employee data to understand demographics and hiring trends. I just describe what I need, and Copilot will automatically analyze my data and create a new report based on my needs. Immediately, I have a beautiful Power BI report that has been automatically created and is fully interactive. I can start slicing and dicing my data to explore deeper.

Copilot added the charts and slicers I asked for, but also conveniently gives me options to adjust the output to get just what I need. For example, let's switch to ask for metrics and trends in a slightly different layout. Automatically, Copilot updates the report. Let's go ahead and keep this report page.

It goes much further. I can also ask questions such as why is our attrition rate going up? Copilot responds by adding a new page to my report and tapping into Power BI's built-in advanced analysis capabilities for finding key influencers for variables in my data. In addition to giving me a summary of an insight about different employee types having different attrition rates, I can easily explore Power BI's key influencer visual to see what's driving the biggest impact in our attrition rate. I can even automatically see the most significant segments in my data just by clicking.

Now let's go back to the first page and finish up our new report. I'm going to ask Copilot to make it look like our existing exec dashboard. And instantly, it's applied the same formatting and style, and adjusted the layout to match.

Best of all, I can still interact directly with the Power BI report that Copilot has created. For example, if I wanted to manually change this bar chart to a tree map on my own, it's easy to do with just a few clicks. Let's add employee functional area, and now I can filter by this new field.

Finally, to make the report even easier for my team to understand, I'm going to ask Copilot to add a rich text description of my data right inside the report. This narrative summary is fully dynamic, and not only does Copilot highlight interesting insights from my data, it will update the summary every time the data is refreshed, or people filter the report. And just like that, in seconds, I've created a report that would have taken hours or days to do manually.

Copilot built on top of Azure OpenAI is truly revolutionary for how we empower everyone to find and share insights.

(End video segment.) (Applause.)

SCOTT GUTHRIE: And what you saw there is how we're able to organize data in Microsoft Fabric, and in turn, enable this amazing natural language experience in Power BI on top of it.

Microsoft Fabric is designed for the era of AI, and it's serverless data management engine is tuned for advanced scenarios like that.

And with Microsoft Fabric, we have a deep commitment as well to open formats and APIs. Fabric comes with a SaaS, multicloud data lake, which we call OneLake, that's built in and automatically available to every Microsoft tenant. OneLake is built on open formats, including Delta and Parquet, meaning it's compatible with solutions like Databricks and other open source tools.

And since we know that many customers want to build AI solutions using data that might already be stored in other clouds, we've also built Microsoft Fabric to work across clouds, including AWS, and in the future, Google. And I'm incredibly pleased to announce that Microsoft Fabric is now available in public preview for everyone to use. (Applause.) We have some great sessions on Microsoft Fabric, and you're going to be able to learn a lot more about it later this week at Build.

We have an exciting future ahead of us. Azure and the Microsoft Cloud is the place for every developer and every organization on the planet to be able to innovate using AI. We're really looking forward to building some great AI solutions with you together, and I hope you have a great rest of the Build and a great rest of the day. Thank you very much. (Applause.)

END