Chapter 6

Tech Trends

Emerging Technologies: The Very Latest

Welcome to Tech Trends

(mellow electronic music) - If you've been a software developer or worked in information technology for any length of time, you've seen hundreds if not thousands of products and technologies come and go. Some make huge long-term impacts on how we work and play, while others fizzle out. When a technology first appears, it can be difficult to know which of these paths it's going to take. In this course, LinkedIn Learning's Technology Staff Instructor Team shares their insights about products and technologies that have just appeared on the horizon. What are these new tools? Why are they potentially important? If there are controversies, what's being said on each side? And as a technologist, what do you need to do to learn more and prepare for how they may affect the way you work? (mellow electronic music) I'm David Gassner, Managing Staff Instructor for LinkedIn Learning's Technology Library, and I'd like to welcome you to Tech Trends.

System outages: Recovery and resilience

- How do we prevent our computer infrastructure from collapsing due to a software update or human error? And can we totally prevent this from happening? Regional and global service and software outages happen more often than we'd like, and when they do, we're reminded of how much our society depends on our sometimes fragile digital infrastructure. In July, 2024, millions of computers went down worldwide forcing banks and airlines and hospitals and other critical infrastructure to halt their operations as their computer systems were manually reset. The cause? An automatic update from CrowdStrike, a company providing critical enterprise security software. In 2022, an outage from Canadian cell phone provider Rogers affected millions of users across the country. The cause? An error introduced during routine maintenance. And the same thing happened with Meta in 2021. During routine maintenance, a misconfiguration of a system took most of their platforms offline, causing a global outage. And there have been several highly publicized near misses, including Log4J, Shellshock, Heartbleed, and the notorious Y2K bug. All narrowly prevented thanks to global efforts from system administrators. Seeing these outages happen and seeing their widespread consequences, there's some questions worth asking about how to build more robust and resilient processes to prevent outages when we can, and how to get our systems back online as quickly as possible when they go down. To bring some clarity to this, I reached out to a handful of our LinkedIn learning instructors for their insight. To start, I asked what service providers can do to prevent these outages from happening in the first place? Here's cybersecurity expert Tia Hopkins. - Some of the low hanging fruit is to to do rollouts in stages. You know, global rollouts all at once, if they go well, they go well. If they do not go well, it can be catastrophic, as we've seen. Have rollback plans in place just in case things don't go well. - Now, roll backs aren't always an option, especially if the update is to a core system. So testing before rollout is equally essential. Cybersecurity expert and author Caroline Wong has more to say about this. - It's critically important for service providers to conduct security testing as well as quality assurance testing on their updates before those are pushed out to their customers. - Because these updates are often critical, there's a tension here between testing to make sure everything is perfect and rolling out updates to protect users. TIA explains. - The balance that has to be striked for a vendor of security software is that balance between, Hey, we've got an update here that's going to protect our customer base and the longer we delay the rollout of this update, the more our customers are unprotected from this thing. - In other words, providers can do everything in their power to get it right on their end, and even then there's still a chance of something going wrong. Which brings us to the next question. What can IT and security admins do to reduce the risk of their systems going down when something does go wrong during an external service update or routine maintenance? Strategic cybersecurity leader, Mike Wylie, starts out with a simple message. - The best thing IT admins can do to reduce the risk of their systems going down is to partner with a vendor they trust and that has a strong track record. - With that trusted vendor relationship in place, Caroline and Tia point out the necessity of understanding your own systems and having robust policies and practices in place. - Any organization is likely to have a unique or specific setup with regards to their tech stack. They might have different versions of different software components, and it's important for IT admins to test this on a few systems before deploying across an entire production set of systems. - Having strong rollback plans in place is important. Having a sandbox environment where updates can be tested before they're rolled out globally, so staged rollouts, QA testing, you know, testing different types of machines with different configurations before updates are rolled out, and ensuring change management protocols are clear and understood. - Which is all well and good until something goes wrong. Here's Tia again. - This is a huge realization of how much we rely on technology. What will you do if your options for recovery do not include the technologies that you're used to leveraging? - That's a great question, and part of the answer, according to Caroline, is to pay close attention to the risk level of everything that happens in your systems. - It is critical for organizations to determine what are high risk actions? For example, a high risk action is going to be when a software provider pushes a mandatory update. This needs to go through extra rigorous QA and security testing. - This begs a question I've heard often and even grappled with myself. If I know an automatic update may be risky, should I just opt out and stop it from taking place at all? Tia points out the complexities of this dilemma. - On the one hand, you could allow, you know, automatic updates and experience what we just experienced with this global outage. But on the other hand, you could disable automatic updates and neglect to install an update that would protect you from something and then have a breach. - Which brings us to today and why we're here talking about this. With so many different systems and stakeholders connected together in our information infrastructure, is it at all possible to build truly robust systems that are to these type of outages? Mike rounds us out with a reality check - To build a fully resilient and redundant system, there would be extreme cost, complexity and likely would introduce new vulnerabilities that result in a worse outcome. - Did you catch that last part? Trying to build a perfectly robust system might introduce new vulnerabilities. I think this is the thread running through this entire conversation. Cybersecurity is not about perfection, it's about continuous improvement in a rapidly changing landscape. When we experience these massive system outages, it's usually from a new novel cause. And whenever they happen, cybersecurity experts like Caroline and Mike and Tia dive in to figure out what happened, how to fix it, and most importantly, how to stop it from happening again. So here are my takeaways. Number one, testing at every stage of the process is essential. Number two, phased rollouts ensure errors can be caught before they go global. Number three, critical systems need rollbacks and redundancies for when things go wrong. And finally, number four, in place of chasing a utopian robust system. We work together to find issues, solve them quickly and learn from them. That's what we already do and that's what gives us the resilience to recover and build back better when our digital infrastructure goes down.

NPUs vs. GPUs vs. CPUs

- Computing got a major upgrade, and it's because AI is revolutionizing processing with the integration of a unique chip, the neural processing unit. Let me explain. There's been several shifts in computing architectures over time, starting with the humble CPU. The CPU is the brain of every computer, translating instructions from higher level languages into machine code that hardware can understand, while also managing interactions between systems in the computer. Over time, CPUs rapidly gained features and capabilities like multiple cores, faster clock speeds, and improve power efficiency. With the rise of 3D and computer gaming, the need for specialized processors developed. That's when GPUs, or graphic processing units, appeared as special assistants to the CPUs. The differences between CPUs and GPUs are the types of math problems they're customized to solve, like matrix operations, vector calculations, and floating point arithmetic. Modern CPUs often have 4 to 16 cores, while GPUs can have thousands. Although each GPU core is less powerful than a CPU, some features of GPUs like their abilities to handle massive amounts of parallel processing made AI engineers regard GPUs with envious eyes. So they drew their plans to adopt GPUs as the primary way to process AI tasks, and that worked for the most part. Eventually, just as GPUs were built to offload the graphic tasks they excelled at, neural processing units excel at AI calculations like matrix multiplication, low latency, and high throughputs. Google calls their NPUs tensor processing units, and Apple calls them a neural engine, while Microsoft recently announced Copilot+ PCs that can handle 45 trillion operations per second. But there's another reason why NPUs are fundamental to AI computing, and that's the rise of edge computing. So far, models like GPT or Gemini process prompts in the cloud, which can be expensive and expose information on your local machine to the internet. With NPUs, your local machine can run a small language model like Phi or Gemma. Although these are not as capable as large language models, they can do a lot of work while having access to your local machine's context. The small language models can offload more complex tasks to the cloud when necessary while keeping your private information private. As a bonus, with this new architecture, AI can become more efficient, saving energy, and providing greater capabilities. That's a win-win for developers as well as the future of computing. What will they think of next?

New Google Gemini models and Google I/O announcements

- Google I/O brought a number of features focused on an expansion to Google Gemini. Let's start with new features added to Google's chatbot, Gemini Advanced. Google Gemini Pro 1.5, which used to be only available to developers, is now inside their Google Advanced platform that lets you use a model with more capabilities as well as a huge 1 million token context window. You can use Google Docs, PDFs, and Word files from Google Drive or use your own file uploads. I can easily add a large volume of an old encyclopedia and ask a very specific question from that huge context. That mailing token context window means you can look through an hour's worth of video, 11 hours of audio, 30,000 lines of code, or a 700,000 word document. Another huge announcement is a low latency multimodal model called Gemini Flash. It offers advanced reasoning and huge context windows of up to a million tokens. You can see on this chart that it performs pretty well compared to other Gemini models. You can play with it right now in Google AI Studio and Vertex AI. And by the way, you can also sign up for an extended 2 million token context window if you're a developer. This is Google's answer to OpenAI's GPTo announced at OpenAI's Springtime Update a day before Google I/O. Although I have to admit that the model isn't quite as capable, check out the pricing. It's extremely affordable at 35 cents for content below 128,000 tokens and 70 cents for up to a million output tokens, as well as 53 cents per million output tokens for prompts of up to 128,000, and $1.05 cents for longer prompts. Now, this is much cheaper than GPTo, which right now is priced at $5 per million input tokens and $15 per million output tokens. So, this is about the cheapest price for a capable multimodal model. And coming soon, the model will also have something called context caching. This lets you store large documents so that you can reuse them without having to send them every time you want to prompt. Another huge piece of news for developers is the availability of new vision features for Gemma. That's the open source model based on Google Gemini. You can find Gemma and start playing with it today on Vertex AI and other platforms. So far, there's been only two flavors of Gemma, RecurrentGemma and CodeGemma, but starting today, you can use a new version called PaliGemma, a multimodal model with vision capabilities. Gemma has currently been only available in two tiny sizes, 2 billion and 7 billion parameters, but will soon be getting a 27 billion parameter version. Gemini is also starting to roll out on Google Search with real-time information and igentive capabilities. You can try this out today with something called AI Overviews. Google will generate an AI-powered custom Search results page. You can see this today by searching for dining or recipe information, but it's going to be added to movies, hotels, shopping, and a lot more. Let's take a look at a few of the products that were announced, but you can't really use just yet. The first is Project Astra. That's a universal AI agent that continuously processes and responds to whatever it sees in real time through video. Now, this is something that Google demoed last year and just yesterday OpenAI showed something similar. However, don't get too excited, this is going to come to Gemini apps later this year, and we don't even have the OpenAI version. Imagen 3 is the new version of Google AI's image generation model. Although that's not available yet, it's something that you'll want to sign up for in Google Labs, which is a great place to find experimental Google projects. So, take a look around if you want to try some of their other emerging tools. Imagen 3 is supposed to be more realistic, more responsive to prompts with better text rendering. And if you've seen some of the latest high quality generative AI video from products like OpenAI's Sora, VO is Google's version, which can generate impressive longer video. You can also sign up for this by going to Google Labs, but no word on when this is coming out. Now, although there are other smaller announcements, right now pretty much everything else isn't something you can use today. You can assume that Gemini will be rolling out throughout all of Google's products. Their approach is to generate automated agents that will help you get things done quicker. But don't worry, when there's cool new stuff you can actually use, we've got you covered.

GPT-4o, multimodal AI, and more

- OpenAI had their spring update on May 13th, 2024, where they released their latest model named GPT-4o for omni and a bunch of other stuff. Here's the breakdown and what you need to know. The main update is the release of GPT-4o, the first model that integrates text, image and audio that can be used to combine all these three modalities in both the input and the output phase. In effect, this means what used to require multiple operations turn speech into transcription, then run the transcription through GPT, then turn the response back into speech is now simplified and takes significantly less time. The new GPT-4o model is described as being two times faster than GPT-4 Turbo and the GPT-4o API is two times faster, 50% cheaper, and has five times higher rate limits than GPT-4 Turbo. Long story short, GPT-4o replaces GPT-4 Turbo as the new benchmark model from OpenAI today. With this in mind, here are the four things you need to know. Number one, ChatGPT with GPT-4o is now free for everyone. With the release of GPT-4o, OpenAI also opens the gate for everyone to use the latest and most powerful model on ChatGPT, even those without an account. All users, including free users, now have full access to GPTs from the GPT Store, vision, web browsing, memory, and advanced data analysis. Premium, Team and Enterprise users also get better performance, higher use limits, so 80 messages every three hours, and earlier access to new features. When free and premium users exceed their usage limits, ChatGPT will revert back to GPT-3.5 Turbo, as before. Number two, multimodal is rapidly becoming the default. Our science fiction dream of an omniscient voice-controlled AI assistant gets ever closer. And over the next few weeks, OpenAI will roll out full live voice and vision capabilities for ChatGPT, meaning you can now have more fluid conversations with the app and show it things through the device camera and get responses in real time. New to the audio model is significantly reduced lag, leading to more natural conversations, the ability to interrupt the model in mid-sentence and mid-reasoning, and having the model attempt to identify and respond to your emotional state with a similar and appropriate emotional tone. The voice model is also coupled with an improved live vision model where you can ask ChatGPT about what the camera is seeing in real time and get voice feedback. In the demo during the launch, the team had ChatGPT lead them through a basic math problem and commenting in real time on the math as the user wrote it out on a piece of paper. This hints at an immediate future where AI assistants not only respond to prompts, but can be set up to actively take part in solving tasks. Number three, to help users take advantage of these multimodal features, OpenAI is releasing a ChatGPT desktop app with new integration features. The app is macOS only as of this recording and features the standard ChatGPT voice mode and image upload on release. GPT-4o's new voice and video features will come in the app in the future, according to OpenAI. Using the desktop app, you can now talk to ChatGPT directly without opening your browser or device and ask it to look at screenshots from your desktop to do things like help solve a coding problem or provide information and feedback on an image or analyze a graph. And all of this brings us to number four: new security implications. Broader availability, new input and output modalities and ChatGPT as a native desktop app highlights existing security issues and introduces new ones, especially for the enterprise. As ChatGPT becomes easier to use through its voice and vision capabilities, users are likely to perceive the app more and more as a natural collaboration partner, and once they have it as a native desktop app they can talk to and share their screen with, policies and guardrails around when and how to use AI assistants in your work become paramount. Put in plain English, there's a huge UX difference between copying and pasting or taking a picture of code or information to be processed by third-party app and just clicking a button in an app and ask What's wrong with this code? Or help me understand the spreadsheet. Bottom line, with the new capabilities of GPT-4o comes a heightened urgency for robust policies, practices, and oversights when it comes to AI use in any privacy and security-oriented environment. OpenAI has been the de-facto leader in the generative AI space since the release of ChatGPT in November 2022. This release of GPT-4o, combined with the unlocking of ChatGPT, shows the company pushing themselves and the entire leader pack towards the future where multimodal conversational AI with voice and vision input is front and center. This is in line with what all the AI companies are doing right now, and I expect we'll see this type of multimodal interface become the new standard for our AI interactions very soon. The future is here today and it's multimodal.

data build tool (dbt)

(upbeat electronic music) - All right, picture this. A rapidly growing company is seeing more and more demand for its products. While the front-facing business thrives, the vast and varying data behind the scenes paints a picture of increasing complexity for its data analysts. Now, this company, like many others, recognizes the need for a seamless, efficient way to unify and analyze their data. Their mission isn't just to understand what's growing their business, but also to understand and evolve from within. In short, improving their data processes and insights to keep up with the core business. That's where DBT gets involved. DBT or the Data Build Tool is a tool that enables data analysts and engineers to transform and model data in the data warehouse. If SQL is the language to do this, then DBT is the framework around it that adds structure, testing, and documentation. It's best to use when teams want to treat SQL like software, incorporating practices like version control, automate data testing and documentation, or make their analytics pipeline more modular and collaborative. A data pipeline with DBT can vary from company to company, but imagine the pipeline as a conveyor belt. At the start, raw data is ingested into the data warehouse. With traditional ETR processes, you'd transform this data elsewhere and then load it, hence extract, transform, then load. But with DBT, you follow an ELT approach. The raw data is loaded first and then transformed within the warehouse. Extract, load, then transform. DBT helps you create, test, and document these transformations, ensuring the end data models are accurate, up to date, and ready for analysis. I'll also call out the DBT website, which has a wealth of use cases that puts a lot more detail and examples on how varied these pipelines can be. At its heart, DBT Core is an open source offering. It provides the foundational features, letting you run, test, and document your SQL transformations from the command line. On the other hand, DBT Cloud is a managed service built on top of DBT Core. It offers everything Core does plus a web interface, schedule runs, and more collaborative features. Think of DBT Core as the engine while DBT Cloud is the car built around the engine, providing additional comfort and functionality. To get started with DBT, I always encourage everyone to try something hands-on to get a good grasp of what it is and how it works, but alongside that, diving into DBT requires a specific skillset. Understanding SQL is essential. It's the heart of DBT, and ultimately, DBT is compiling SQL for you, so you'll need a good graph of SQL in order to help you construct DBT models that take full advantage of the SQL capabilities. Linked to this, data warehousing principles can help inform your high-level approach to transformations and storage of your data for analysis. On the softer side, I often like to link some aspects of data modeling to problem solving. The skills to identify problems, think through them, debug them, and then build solutions around them is quite important. Often you won't be doing this alone, either, so you're likely building models for others to use, so collaboration will be another soft skill to call out here. And finally, data engineering is a fairly new role. It's a high level sort of set of skills, but ultimately the field is still evolving quickly and so you want to be continuously learning and keeping in touch with what's new and how businesses are using DBT to solve challenges.

Microsoft Dev Box

(energetic ambient music) - Ask any programmer to describe their ideal development computer. I guarantee that every person will give you a different hardware and software wishlist. The time to create a new computer that meets the developer and company requirements can vary with traditional deployment methods. The IT team might take days to provision the new box cloud deployed computers speed up this process. Microsoft built their dev box service to address this trend. It's part of Azure, fits into your Microsoft IT management tools and supports a variety of configurations. Plus it's fast provision. You can have a workstation up and running in minutes. Microsoft Dev Box is a service that lets you create and use high performance cloud-based workstations that are ready for coding. Your developers have the freedom to create their workstations in a self-service environment. They can choose from a selection of pre-configured dev boxes which contain curated images prepared by their development team. These images come equipped with all of the necessary development tools, the code repositories and the software development kits. As a result developers can start working immediately as soon as their chosen dev box is deployed. These dev boxes are indeed easy to provision but equally important, their management and security are ensured through centralized governance tools like Microsoft Intune and Microsoft Endpoint Manager. This means that regardless of where the dev boxes are operating, IT administrators and infrastructure teams have comprehensive and centralized device management and governance tools in place to track and closely monitor each deployed dev box. This oversight guarantees that security, compliance and cost management are maintained throughout the system providing a seamless and secure working environment for the developers and the organization. You can read more about Microsoft Dev Box on this Microsoft site. Let's spend a minute talking about how to work in Azure to create dev boxes for a project. Microsoft built the system so that IT administrators, platform engineers and developer leads can collaborate to do this. Here on the site, you can see this graphic that talks about the steps that you need to do to create a dev box and this example on Microsoft site. The steps one through six are implemented by the platform engineer, and then all the developer needs to do is create the dev box. For example the platform engineer might create a dev center. This is where they organize their project resources. Next state configure network components to enable the dev boxes to connect to organizational resources. For example, joining the box to an active directory or setting up virtual networks. Next, someone defines the dev box definition. It's here where they set the source image, compute size and storage size. The next step is to create a project. After that, you'll create a dev box pool. Think of a pool as a collection of dev boxes that are managed together and have similar settings. You can create multiple dev box pools to support the needs of hybrid teams that work in different regions or on different workloads. Finally, you'll add some users to the dev box project. At this point, any user in the project can create a dev box. Dev box users can log into Azure then create dev boxes from any of the pools and manage those dev boxes from the Azure developer portal. What can you do with your dev boxes? You can work on multiple projects or tasks. By creating different dev boxes, you can collaborate with your team members using shared dev boxes. You can scale up or down your dev boxes depending on your workload. Dev box also helps you manage and secure your developer workstations. You can control who has access to your dev boxes and what they can do with them. Keep your dev boxes up to date with the latest patches and updates. You can also isolate and replace compromised dev boxes quickly and easily. Dev Box is an Azure service that works with your existing Microsoft 365 or Windows licenses. You only pay for the compute and storage resources that you use. To be clear there are other ways to accomplish rapid provisioning and configuration of developer computers. Some teams use virtual machines or Docker containers for this purpose, Microsoft is not alone in this space. Other vendors have similar offerings like AWS Cloud 9, Jetpack Dev Box, or JetBrains space. The existence of multiple options in this space indicates that there is an ongoing search for efficient ways to quickly prepare developer workstations. In this context the Microsoft Dev box presents a compelling option mainly for companies and teams that already rely on Microsoft IT tools or utilize Azure services. Its integration with existing Microsoft infrastructure makes them that attractive choice for those seeking a seamless and familiar development environment.

OpenAI API

- For AI systems to be truly useful in a development setting, they have to be fully configurable and provide basic features like state management and dynamic controls. But when OpenAI first released their API, developers had to do all this manually. So when they built custom agents, they had to declare system assistant and user messages with every request and manage their state by storing each message response pair in some form of database, and then pass that to store data back into the API with every message. It was labor intensive and expensive and quite messy. In response to this, OpenAI has now introduced the Assistants API. This new API is the developer version of ChatGPT's Custom GPTs, and with it, you can create custom assistants with their own instructions, their own tools, and their own knowledge retrieval and spin up individual stateful chats under each assistant without having to store the exchange on your end. This makes developing custom agents with API significantly easier and opens the door to far more rapid and advanced AI development. You can create and configure assistants programmatically through the API and also through the OpenAI Playground, and you can change them at any time through both of these modalities. Now, assistants work by tying three distinct components together to create a seamless user experience. First, there's the assistant itself. It has a unique ID and you can create as many assistants as you like. Each assistant holds the main system instruction or prompt, configuration of tools, including custom function calls, the code interpreter, and content retrieval, and the ability to upload knowledge. So, documents to be used by the assistants for reference. Next, there are threads. Each thread is a separate continuous chat with an assistant, with its own unique ID, and you can have as many threads as you want with any one of your assistants. The thread stores the chat history, and this is what makes each chat stateful. And this means you can have multiple parallel threads and jump back and forth between them based on their ID without losing the thread. I mean, losing the state of each separate chat. And finally, there are runs. Each run contains a message or prompt sent to the API and the response from the AI. Think of a run as a prompt-response pair. The interesting thing about runs is they can have multiple steps, each of which is executed separately. So for example, when you send a message, the assistant may trigger the code interpreter tool and a custom function call for responding, totaling three steps. This means a single prompt can do multiple things at once, even fire off sequences of functions and tool calls. Now, one important thing to note here is the way assistants preserve states and context for each of these chats is by passing the entire thread back in with each run and then adding the new message and response to that thread. So as you go back and forth with the assistant, the token count for each run will grow as long as the conversation keeps growing and can end up consuming a lot of tokens. Bottom line, with assistants, you can now programmatically spin up custom assistants in the GPT environment and set them up to do whatever you want and integrate them in any way you want with your own applications.

AI pair programming

(Upbeat music) - If you write any kind of code chances are in the near future you'll be working with an AI pair programmer. If you're not doing it already, you've probably heard people talk about or even showed demos of how they use ChatGPT for programming. Give the AI a basic prompt, like create a React JS component for displaying responsive images and out comes fully written code you can just cut and paste into your project. So let's talk about this idea of AI as a pair programmer, what they can do, how they work behind the scenes and what we can expect from them in the near future. There are many AI pair programming tools available already and I'm going to focus on three of them because they provide a good framework for understanding how all of this works and are good representations of the different categories of tools available. They are ChatGPT, the general purpose large language model, Microsoft Bing which uses ChatGPT as an interface between you the human and the internet, and GitHub Copilot, a specialized AI pair programmer embedded in your code editor. First up, what can an AI pair programmer do? The word generative is the key phrase here. All these AI tools can generate code for you based on prompts but that's a short step into this possibility space. AI tools can also do things like detect errors in your code, explain what a string of code does, and add comments to it. Even reformat code to your specifications, translate code from one language to another and write tests for you. And specialized tools like GitHub Copilot don't even need prompting. They effectively work as supercharged intelligent auto complete, finishing your sentences and even thoughts based on comments and the code you write. In short, these tools help you write code faster, more efficiently, and saves you time by providing completed code examples, patterns and prototypes when you need them. Which begs the next question. How does this work? How does an AI know how to write code? The simple answer is code is just language and large language models like ChatGPT are designed to generate sentences emulating language based on its training data. The long answer is AI companies have trained large language models using examples of coding languages and as a result, these AI systems are able to generate sentences emulating those coding languages. Now, it's important to understand there is no actual intelligence, no conscious thinking agent inside these systems sitting there writing the code. What's really happening is generative AI interprets your prompt and responds by putting together sentences word for word based on statistically guessing at what the next word in the sentence should be. And since coding languages are far more structured and patterned and logical and model based than our regular human languages the chances are the AI successfully putting together functional code is quite high. That said, because these AIs are not actually writing or testing code, but instead putting together sentences that look like code, in many cases, they're just repeating the existing coding patterns found in their training data. As a result, AIs have a tendency towards repeating old code patterns and standards. This is where we start to see the differences between different AI programming tools. ChatGPT and other large language models are meant to be interfaces between humans and data. If you input data, they can work with that data to do things like summarize an article or translate code from one language to another. But if you ask ChatGPT to generate code from scratch, it'll fall back on its existing training data, which as I record, this video is capped off in 2021. So if you ask it to generate code based on a standard introduced later, like for example CSS layers, it has no idea what you're talking about and will provide an incorrect answer. Bing is different. It provides generative AI as an interface layer between you, the user, and the internet. So when you ask Bing to generate code from scratch or explain how a new standard like CSS layers works, Bing sources the internet for authoritative information before providing the answer and includes links to sources in its response. That way you can check its work, go to the original sources and check out further information. GitHub Copilot is, again, different in different ways. It's very much like having another programmer working with you inside your code editor suggesting code based on your comments and trying to auto-complete your code as you write it. And when you get curious about where a specific code suggestion comes from, you can open a Copilot panel and see the source material in real time. You can even select one of the original sources as your inserted code if you want to. GitHub is also working on additional tooling for Copilot to provide features like code explainers, translators, and transformers even test generation with one click. So what have we learned here? First of all, AI can be a huge help for programmers when used correctly and these tools are going to get better fast and keep getting better as we use them. Secondly, AI programming tools are not going to take jobs away from human programmers anytime soon. You still need a skilled human in the mix to get anywhere beyond the basics. That said, these tools save you a lot of time both in doing online lookups and in writing the code itself. Now, based on all this, what is my recommendation to you on how to use these tools right now and in the immediate future? First and foremost, incorporate GitHub Copilot or a similar tool in your process right now. They truly are intelligent, supercharged, auto complete and they make writing code a lot faster and easier. I expect we'll see rapid evolution of these tools in the near future. So now is the time to get acquainted with them. Second, if you choose to use a general purpose AI like ChatGPT, for things like online searches use one that is actually connected to the internet and provides references for its output. In other words, while ChatGPT is the hot topic right now, Bing is a better choice simply because it actually sources the web and tells you where it got its information from. Third, think of all of these AI coding tools as overconfident junior developers with lots of historical knowledge and a shallow awareness of modern standards. By their nature, they tend towards repeating prevalent patterns and those patterns will, by their nature, be old. Programming languages, on the other hand evolve much faster than their real world implementations. So you still need to be on top of modern standards and best practices so you can coax the AI along the right path. Bottom line, pair programming with an AI is already becoming the new norm and will be a huge help for you whatever level you're at. So jump in and start training your own AI pair programmer today.

GPT-4

- I remember my very first game controller. A simple joystick from the Atari 2600 with just one button. Now, at the time I thought it could do everything, but later we got controllers with more buttons, D pads, and fancy new features. Every time the upgrades felt like a big improvement. And that reminds me of the upgrade we just got for the fastest growing technology app of all time, GPT version four. And this update is huge. So what is GPT? It stands for a generative pre-trained transformer and it's a neural network that creates new content, like stories or art, and is pre-trained using massive amounts of data. A transformer means it understands key points from human input and tries to infer your intent. Let's say you ask a question like, how do I build the best house in Minecraft? The transformer identifies keywords like build, best house, and Minecraft, but since you didn't provide details like the version of the game, materials available, or that you happen to be in the snow biome, it will make assumptions and give you an answer that is the most likely to be correct for most situations in the game. So what's the difference between GPT-3 and GPT-4? The new model is better at reasoning as well as providing concise answers, although it's a bit slower than older models. That shows up in how well it performs tasks, but also means that it requires more computational power which can be more expensive. OpenAI, the company behind GPT, has worked to fine tune the model to reduce so-called hallucinations. In other words, produce answers that are nonsensical or untruthful. They have also attempted to reduce biases and improve the overall safety of the results. The model is currently available through open AI's ChatGPT Plus subscription and is also part of Microsoft's Bing as well as Microsoft 365 and Azure. You may have heard that GPT was able to pass the bar exam, which lawyers have to take. While GPT-3 managed to pass this test within the lower 10th percentile, GPT-4's performance was in the top 10th percentile. GPT-4, like GPT-3, was trained on data from before September, 2021. But unlike GPT-3, GPT-4 is capable of using web links as input and can accept much larger amounts of information. It can accept inputs of up to 25,000 words of text and it can provide longer outputs as well. So for example, you can provide the transcript of a long conference talk or a livestream and ask GPT-4 to summarize it and give you the key topics with time codes. Another key feature is steerability. You have better control about the personality, verbosity, or style. You can, for example, ask GPT to put itself in Socratic mode so that it answers questions by asking leading questions, which allows students to slowly uncover the answers themselves. One of the exciting new features is the ability of GPT to understand photos and graphics. So you could, for example, take a photo of the inside of your fridge and ask it to give you a recipe based on what you have. You can pass it a screenshot of a chart or graph and it can accurately interpret the context of the data and do calculations based on that information. This ability is not currently available to the public, but it will be rolled out over time. In the demo from OpenAI, one developer took a photo of a simple sketch and GPT wrote the HTML and JavaScript to build an interactive webpage. GPT-4 is being added into all aspects of life. ChatGPT has already become the fastest growing product in the history of technology with 100 million users in the first two months. And you may have heard of GitHub's co-pilot, which has helped developers become 88% more productive. Companies like Duolingo, Stripe, and Morgan Stanley have already added GPT-4 implementation into their products. Recently, Microsoft announced the Semantic Kernel, an open source SDK that will make it easier for developers to integrate GPT and other AI features into all applications. Its features are being rolled into Microsoft 365, so apps like Excel, Word, and PowerPoint will eventually see productivity gains that have been available to developers in products like Co-Pilot. GPT-4 is like the best controller upgrade for AI language models. And it's going to change the way we interact with our world as developers begin to add these capabilities into their applications. So get ready to experience new features, image inputs, more accuracy, and a whole new level of creativity. Coming soon to an app near you.

Copilot for Business 1.0

- GitHub Copilot got a big upgrade since its release in June, 2022, and it includes a version for business. Here's what you need to know. Since the release, we've learned a lot more about what developers have been doing with Copilot. According to GitHub's research and usage data, Copilot has been generating 46% of developers' code across all programming languages, and 61% for Java developers. And in terms of satisfaction, 90% of developers are reporting they complete tasks faster with Copilot, with 75% reporting a better ability to focus on more satisfying work. Which is the whole purpose behind Copilot. In addition to that, GitHub has improved the OpenAI Codex, the artificial intelligence model that parses natural language into code. Copilot used to look at the code that came before the insertion point to make suggestions. The new codex uses a model called Fill In the Middle or FIM, which also looks at the code after the insertion point so that it has more context about what you're trying to write. There's also a new lightweight client-side model that keeps track of your preferences and behavior as you use Copilot. So it knows when you've accepted suggestions and it tries to be more accurate about what it serves. Copilot is also a lot better at preventing the suggestion of insecure code to users. GitHub is also now available for business which adds enterprise features like license and policy management. Plus, proxy and corporate VPN support. The business plan is almost double the price of the individual plan at $19 per seat. But companies can choose to use Copilot without having to store their code on GitHub, and developers can integrate their code with other editors like JetBrains Visual Studio and Neovim.

ChatGPT

- This semester, my daughter started a class in college and got a note about the policy for ChatGPT use and that got me thinking that AI and machine learning are already permeating our culture. Now ChatGPT is an online application, which launched in November 2022. It reached one million users in only five days. Now the application lets users have a conversation with a set of technologies known as GPT. GPT stands for generative pre-trained transformer. Its goal is to generate or create new content. In order to do this, the tool is pre-trained with up to 175 billion parameters. To process this data, it uses what developers call a transformer, which is really great at understanding the way people write sentences. ChatGPT was created by OpenAI, a company with a mission to create something called AGIs or artificial general intelligences. An AGI could understand any intellectual task that a human being can do. OpenAI's goal is to create AIs that benefit humanity, not replace it. You may have heard about some of their other products. DALL·E 2, an AI system for creating realistic art, as well as Whisper, a speech recognition, translation and transcription system that approaches human-like synthetic speech. You may be wondering how computers are able to do things like creative tasks, that up to now, only humans could do. Now the answer is that developers have created models and algorithms that attempt to duplicate how humans solve creative problems. Here's an example. I worked at a newspaper and I would watch artists doing portraits. They would print out reference material which included photos of the person from different angles and perhaps styles that inspired them. Then they would translate that inspiration into an original portrait. Now to most people, that process seems magical, but developers recognize that sequence of events as an algorithm, a repeatable set of steps that yields predictable results. As the name implies, ChatGPT is focused on conversations and it's capable of writing essays, scripts, resumes, songs and even more accomplished tasks like writing and debugging code. In the case of GPT, the model is known as autoregressive, which means that it's trying to predict results based on past values. That's why ChatGPT seems human at times. It tries to predict not just what you asked for but also what you probably mean. ChatGPT does have some limitations. It can deliver answers that are wrong and do so authoritatively. It was strained mostly with data based on 2021 and earlier, so it doesn't know a lot about current events. Now this version was also trained using human feedback to fine-tune the model so that it was better than its predecessor. It's considered to be version 3.5 of GPT. It's also able to replicate the mechanics of generating content, but it can't understand whether that content is subjectively good or bad. In the future, I think it's going to speed up the development of tasks by helping humans iterate on ideas. ChatGPT, like other artificial intelligences, is more than just an application. There are APIs available that allow you to create your own products based on the models. ChatGPT itself is just an application based on the GPT API. Now the biggest opportunities in this space are for developers that can understand and use the APIs, entrepreneurs that can leverage the technologies to create new and exciting products and for professionals who understand how to work with AIs and get the most out of what the tools can do. Like I heard someone say, you're probably not going to be replaced by ChatGPT, but you might be replaced by someone who knows how to use it.

The Merge

- What is the Ethereum merge and how does it impact you? In September 2022, Ethereum one of the biggest blockchains, cryptocurrencies, and the platform on which the thing people refer to as Web3 is being built, underwent a major technological shift. The underlying consensus model of the blockchain was shifted from proof of work, to proof of stake. Now, for most people, that's just words. So let me explain what's going on here. Starting with how a blockchain works. The blocks on a blockchain are immutable containers, sort of like safes, where we can put our transactions. So when new transactions are added, they're put into a block. The block is locked and added to the chain. And now if anyone goes back and tries to alter the entries in this locked block, an alarm goes off and we know that something's gone wrong. That's how the blockchain becomes immutable. So how is this different from how we do traditional banking and finance? The blockchain is decentralized and trustless. What does that mean? Consider how a regular bank works. You go to the bank and give them your money, then they hold your money, and then you can make transactions with that money. The bank becomes the single source of truth about how much money you have and where that money is traveling and they have a single ledger where all that is happening, what's called your bank account. Cryptocurrency Blockchains are completely different. Here, in place of having a bank, a third party who is a central source of truth, everyone has a copy of the ledger and anytime a transaction is made, we go and compare each other's ledgers to make sure they all agree and then make the transactions. But this opens the door for a very tricky problem known as a double spend problem. Say for example, I owe you two coins and I also owe Sam two coins, but I only have two coins. Normally I would only be able to pay one of you with my two coins. But in a decentralized system, there's a way for me, to try at least, to scam one of you. You see, if I go to you and say, I have two coins, and I want to pay you, you can cross reference my ledger on your own and see, yes, you have two coins and then I can pay you that. But then I can take advantage of the delay in the system to hurry over to Sam and say, "Hey, check your ledger and mine. And you see, I still have two coins. Okay, I'm giving them to you too." Thus I can spend the same two coins twice, the double spend problem. This is a problem unique to decentralization, and it's also one of the reasons why decentralization is so hard. If you have many different sources of truth, how do you ensure they're all in agreement on what is true and what is not? The solution created by Bitcoin and then propagated through all crypto projects, was to introduce a consensus mechanism. Basically incentivize users of the network to validate only proper transactions on the blockchain and add them into blocks by paying them in coin. So, the cryptocurrencies you hear about on all of these blockchains are the payments made to the validators of the transactions that sit in the blocks on the blockchain. You are paid to maintain the blockchain, and now we're finally getting to the merge and this transition from proof of work to proof of stake. You've probably heard stories of how Bitcoin mining uses as much energy as a small country. This is because Bitcoin uses the proof of work model and up until the merge, Ethereum also used the proof of work model. Proof of work makes it expensive to add new blocks to the blockchain and ideas, because it's so expensive in the form of burning electricity to do this, only the people who have valid blocks to add will add blocks to the blockchain because that's the only way they'll get their money back in the form of crypto. The actual validation mechanism is as trivial as it is ingenious. All the validators go and guess at a number. And whoever guesses the right number gets to add a block to the chain and gets paid in crypto. Now, of course, it's a bit more complicated than that, but that's really all you need to know. Computers are guessing at a random number, and the more computer power you have to guess at a random number, the higher the likelihood of you guessing the right number, which is why we get these massive stacks of server base that just guess that number and use a lot of energy to win Bitcoin. To try to solve this problem, a new consensus model was introduced called proof of stake. Here, instead of guessing at a random number, the people who want to validate blocks on the chain and earn cryptocurrency can stake. literally put their crypto in escrow on the blockchain and say, "I'm going to put my money here and prove that I am a trustworthy character." And then as long as I validate valid blocks I can keep the money on the blockchain. But if I approve invalid blocks or scam transactions, or something like that, I lose my stake. Literally, you prove that you have a stake in the game. So that is the merge, the transition from the very wasteful proof of work model to the much less wasteful, but quite exclusionary proof of stake model. At present, the stake for partaking in validation on Ethereum to forge new blocks, rather than mine them on Bitcoin is 32 east, which is somewhere around $40,000 US dollars, right now, as I record this. The promise of the merge is that the energy footprint of Ethereum will be reduced by up to 99% from the proof of work model. Considering all the attention the environmental footprint of crypto has gotten, that's a big deal, but it doesn't solve all the other underlying problems with crypto. This is very much a technological change from proof of work to proof of stake. Nothing more, nothing less.

Prompt engineering

- I'm Xavier Amatrian. I'm VP of Engineering at AI Product Strategy here at LinkedIn and I lead generative AI initiatives across product and engineering. In the context of generative AI a prompt is simply what you use to communicate with the AI and it can be as simple as a question and as complex as a very complicated construct that includes a lot of different components. So generally speaking a prompt can have any of the following, instructions, a question, input data, and examples. So what is prompt engineering? Prompt engineering is a new discipline that is appearing nowadays, and it's very hard to define because it's being created as we speak but it includes all the necessary components to manage prompts at a certain scale. And as I said before, a prompt can be very complex, right. It can be as simple as a question or it can be a set of examples. It can have instructions. It can have code. It can have all different kinds of templates. And if you need to manage that at scale because you're building a product that has an AI on the backend, that needs to be engineered. All that goes into engineering those prompts at scale, it's what we call prompt engineering. We need to also make sure that we understand the difference between the prompt creation aspect and then all the other engineering components that go with it. One of them is, for example, QAing. I mean, you need to QA prompts. You might have different variations of prompt that at scale you want to understand their quality and you're going to have some feedback loop that involves having maybe humans in the loop evaluating different aspects and different component of the prompt and then feeding that back into the engineering and the design of the prompt itself. Prompt engineering is a component of generative AI and I think it's going to evolve very quickly and very fast. And in fact, it's going to be leveraging itself which that is also something that it's mind boggling, but there's already a few experiments going on with AI prompting an AI, right. That's something you can do. You can prompt a language model so it generates a prompt that then goes into a text-to-image generative AI model to generate an image. So you can imagine that there's going to be AIs talking to themselves, and that's going to be part of the prompt engineering near future. But all of that, again, when needs to be managed at scale as part of a product, there's a lot of constraints like how do we protect, how do you make sure that people don't mess around when talking to the AI and making go off the rails? How do you QA and make sure that the quality is the right one? How do you productionize it and keep it working at a certain scale? I think that is all going to be generating a new discipline, really about the whole managing of those AIs and the way we talk to them. I think generative value is going to be revolutionary and transform our industry and probably our world in many ways.

Artificial Intelligence
Generative AI

(electronic music) - Let's play a game that I think you're pretty familiar with. Take a look at the series of images. Could you predict which item should come next? Of course, it's this one. Now let's try again with a new set of images. Which image do you think should come next? Of course, it's the pumpkin. All right, one last time. Now you probably guessed that it's this one, but have you ever wondered why you're so good at this game? You probably also noticed that these images had a similar theme, Halloween. Your brain stores tons of information about familiar objects, but it also tries to create rules in order to predict what is coming up next. Lately, we've learned that we can not only teach computers to do the same thing, but we can ask them to use that information to create new patterns. We call that generative artificial intelligence. Let's take a look at how it works. Computers are great at detecting patterns in a useful way. So for example, phone cameras are great at recognizing faces and they can be trained to recognize if that face is yours. Now they can do this by looking for patterns in the image. They map out distances in different parts of faces, which is called biometrics. It's easy enough for them to make a prediction as to whether that pattern matches what it knows about you. The breakthrough in generative AI comes when you realize that you can train the computer to create a pattern its trained to recognize. Let's say you train a computer with some pictures of a series of noses. You can ask the computer to analyze the pixels in the image to learn what a nose looks like. With enough data, you can ask the computer to generate a series of new pixels that look like a nose. With enough additional inputs, you can imagine how this could be extrapolated to faces and then entire images. You can go to the site thispersondoesnotexist.com and have a generative AI create random human faces. Now the site claims that 90% of fakes are not recognized by an ordinary person and 50% are not even recognized by an experienced photographer. This type of analysis and generation can then be used on techniques like creating a deep fake where you asked the computer to replace an existing face by training it on a series of faces you provide. It's also not limited to images. Pattern recognition and generation is being used to create original music trained by learning from different music genres. Tools like Compose AI use a technology called GPT-3, which has been trained on billions of learning parameters and can write human-like text. Now that technology has been used by companies like GitHub in a tool called Copilot, which helps programmers write entire functions and cuts down the time it takes to generate code. Although a generative AI is pretty impressive, it does have some limitations. First, it requires massive amounts of data in order to get really good at generating new information. Second, it doesn't always generate desirable results. In the case of something like Copilot, you can't trust that the code it writes will actually work. Now third, it can't create anything new. This technology is only combining information from the patterns it already knows. It's a disruptive technology for sure, but right now it's best at helping humans process large datasets, generating tons of options, and shortening the time it takes to handle repetitive tasks.

GitHub Copilot

(mellow electronic music) - GitHub Copilot is an artificial intelligence tool that writes entire functions for you in many programming languages based on the context of your code. It uses an artificial intelligence technology called GPT3, which stands for Generative Pre-Trained Transformer. It's a natural language processor that uses big data to predict what you will be writing, so they can do things like answer questions or write sentences for you. Copilot uses a new algorithm based on GPT3 called codex, which focuses on software source code. Although it works with many popular programming languages and frameworks, it's especially good with languages that have lots of publicly available code like Python, JavaScript, Ruby, and others. Microsoft, who owns GitHub, made a $1 billion investment in the company that created codex called OpenAI. The code is synthesized from public sources like existing GitHub repos. As more and more people use the tool, it will also train codex and help it provide better solutions. You can activate it by starting to write a function with a meaningful name, and Copilot will attempt to create that function for you. Or you can provide a detailed comment that will be used to query the codex. The code it produces is not just copied from open source. It's brand new code generated by the algorithm, and it's surprisingly accurate. Think of it as the next generation of auto-complete. Although right now it's in its infancy, GitHub Copilot is poised to change the way you code in the future.

Facial recognition

(bright music) - The other day, I got a message from a social media network that someone had posted a picture of me. The service wanted to know if I approved of the posting of this image. Now here's the thing. I don't know this person who posted the image. I've never met this person, I don't know anything about them, yet somehow they had posted a picture of me. The photo was of a teacher in a classroom in a country I've never visited, and at first I couldn't see where in the picture I was supposed to be, but then when I looked closer, in the background behind the teacher was a computer screen where a student was watching one of my courses. The facial recognition algorithm had scanned the photo, found that computer image of my face, correctly tagged that as me, and then contacted me to ask, is this a photo of you and is it okay that it's being posted? This is facial recognition in a nutshell. Computer algorithms powered by the latest technologies, high speed internet, Cloud services, high resolution cameras, machine learning AI, and so on, that identify people based on a photo or video. So how does this all work? Well, it starts with data collection. For years, facial recognition companies have collected every available photo and video online and trained algorithms to find the faces then find recognizable features in those faces and then build models that make it possible to recognize faces from photos and videos. Anytime you or someone else uploads a photo of you to Cloud storage or social media or any public place on the web, it's likely to be snapped up and used by these machines to train the models and improve the algorithmic representation of what you look like. Facial recognition is rapidly becoming ubiquitous. We can unlock our phones and computers by just looking at them, smart doorbells use facial recognition to alert us when someone visits or leaves our homes, malls and other public venues use facial recognition to track customers and offer them tailored services and marketing, and also to reduce crime, government agencies use facial recognition to simplify and speed up identity, schools use facial recognition to monitor students, law enforcement uses facial recognition for crime prevention and investigation.

Basically whenever you're in the public or private sphere, if there's a camera present, there's a good chance facial recognition is hard at work doing its thing behind the scenes. Facial recognition is science-fiction made real and just like in the imagined worlds of science-fiction, this technology has both positive and negative uses and outcomes in the real world. On the positive side, facial recognition can drastically simplify many interactions in our daily life from unlocking your phone or even your door, to verifying your identity when you go to school or an event or even travel. And yes, facial recognition can be a powerful tool for both crime prevention and investigation. However, there's some significant issues associated with this technology. First and foremost, facial recognition is based on machine learning and so-called AI and these tools and models underpinning the technologies are only as good as the data they're trained on. Countless studies and real world situations have uncovered significant racial and other bias in facial recognition systems, sometimes leading to serious negative outcomes for people of underrepresented groups, including black Americans being charged with crimes they did not commit, due to misidentification by commercial facial recognition systems used by law enforcement. Secondly, facial recognition impacts our privacy. Facial recognition training sets the images and videos used to create these algorithms are made up of images lifted from the internet or collected by surveillance cameras. In most cases, without our consent or even knowledge. In other words, even though you probably never agreed to your face being added to a facial recognition system, chances are, it's already in there. And once your face is in the system, getting it removed is all but impossible. Combine this with the ubiquity of facial recognition in public and private spaces and we are headed for an interesting future where someone or some thing will always be able to track our every move. Over time, these systems will improve and be able to tell the difference between me actually being in an image and an image of me being in an image and the industry is working hard to reduce bias and miss identification. But as I explained, the cost of this evolution is privacy, which may not be that important to you if you live in an open and democratic country, but may be vitally important if you are a political dissident in a country undergoing social conflict or political oppression or even war. So here are some things to think about as facial recognition enters the mainstream. If you're a regular consumer using online services and going to public spaces, educate yourself on where facial recognition is being used and how to opt out, if you don't want to be tracked. Most online services, including Cloud services and social media have an option to opt out of facial recognition. If you're a developer working with facial recognition, make privacy by design and explicit informed consent central principles to your craft and product. Educate yourself on the very real problem of AI and machine learning bias and familiarize yourself with existing and upcoming legislation, including GDPR, the Right To Be Forgotten legislation in the EU and CPRA, the California Privacy Rights Act in the US. And finally, if you're a company, an organization, or a government agency looking to implement facial recognition, deploy your implementation in a way that requires informed consent and apply due diligence when picking a vendor. Also make sure you limit the scope of facial recognition to only apply when absolutely necessary and consider privacy and other related legislation. While it is convenient to log onto your phone with your face, and I appreciate social media networks telling me when someone posts photo of me, it is worth asking this question. Is the cost of facial recognition to our privacy and security, a price we are willing to pay? I invite you to join that conversation.

Web3 and Related

Web3

(upbeat music) - There's a good chance, you've heard the terms Web 3.0 or Web 3 used recently, especially in conversations about the blockchain, decentralization, the metaverse and the semantic, or even spatial web. So what is Web 3.0? Why are we talking about it now? And what does it mean for you? Quick history lesson, Web 1.0 refers to the original web where communication float in one direction from primary sources typically corporations, media producers and larger organizations to the visitor. Think new sites, search indexes and online encyclopedias. Web 2.0 refers to the social web where communication flows in many directions and information is created, shared, and maintained by everyone. Think blogs, social media and crowdsourced knowledge platforms. Web 2.0 has been around for a long time and it's still where we are today. Web 3.0 or Web 3 is a broad umbrella term referring to the future web made possible by empowered by a combination of current and emerging technologies. It's been referred to as the human first web and the semantic web and the spatial web. And within these different descriptions, we can get an idea of what Web 3 is all about. Well there is no one true definition of Web 3.0 yet. All the different definitions tend to land on the same key principles and technologies. Web 3 is Open and equitable, allowing everyone to take part and eliminating the need for a non-value adding middleman like social media platforms or search engines. Web 3 is decentralized and trustless, leveraging blockchain technology to create immutable data records, allowing each user to control their own data and trust the data created by others without having to go through a central trust authority. Web 3 is semantic, enabling computers, in particular AI and machine learning to understand data and make meaningful connections without human or authority intervention. Web 3 is platform agnostic, treating data as pure data and allowing the user to choose their own means of data consumption. And Web 3 is spatial, allowing users to interact with data and information through AR and VR, as well as in the real world through integration with materials and objects beyond the screens. The Central premise of Web 3.0 today, is it shifts power from intermediaries like hosting companies and social media platforms to the individual. You own and control access to your own data, the system makes it possible for you to trust person to person and person to company interactions like payments without having to have an intermediary like a bank or a payment portal. And you can access, filter and parse data in whatever way and means are meaningful to you without having to go through a big company. In short, the Web 3.0 concept is a return to the original promise of the web platform. Ubiquitous access for everyone through trustworthy systems, without being exposed to monitoring, surveillance and predatory features like behavior based advertising. Sound a bit like science fiction? Well that's because it is. Current estimates say the first meaningful implementation of Web 3.0 is still 5 to 10 years down the road. So for now at least when people talk about Web 3 or Web 3.0, they're using trending marketing jargon to talk about one possible future.

Metaverse

(upbeat instrumental music) (static buzzing) - Oh, hi there. You may have heard about something called the metaverse, and you may be wondering if it's something that you should learn more about. So let's start by defining what the metaverse is. It's the natural progression of what the internet is today when taken into account what mark Zuckerberg calls, "The feeling of present." It's an immersive sensory experience. Let me explain that to you with one of my favorite games. I love playing ping pong but a ping pong table takes up a lot of room in the real world. It's not just a table, but the room around the able. what an immersive experience can do is make your brain think that it's experiencing a new reality. The other problem with my game of ping pong is that it's hard to find someone who wants to play around the house when I do or perhaps someone that will be challenging to play with but not too challenging. Now that's where the metaverse can step in by giving you a virtual environment that makes you think that you're there. I can play against the robot that challenges me at just the right level. The space I need is minimal, and the computer can match me with players of similar skill level on-demand from all over the world. That virtual environment changes your perception of reality and it can make you feel like you're moving around anywhere in the world or even anywhere that's out this world. It's also the merging of the digital, and physical worlds. Like Satya Nadella said, "The metaverse enables us to embed computing into the real world, and the real world into computing." You can ask the metaverse for example to learn about your physical store. Run layout optimizations. Pricing variant through a set of virtual customers to figure out how to best utilize your space, and maximize profits. It could give you real-time stats when you're watching a game at a stadium, and update your fantasy league. Now, hopefully you can see the possibilities are pretty awesome, but you may be wondering if it's something that you should worry about right now. According to Qualcomm, "Meta sold 10 million Oculus Quest 2 headsets." Meta also announced that their content store had surpassed $1 billion on purchase VR content, and surpassed $51.3 million in monthly revenue in January, 2022. The Metaverse is already here, and having a real business impact primarily in gaming but that's about the change. Microsoft, our parent company, has big plans for the metaverse. You may have heard about their $69 billion purchase of game publisher Activision or their 7.5 billion purchase of Bethesda. This purchase is not just about the games but also the environments, the developers, the code and the user base. This gives 'em access to hugely popular brands like "Call of Duty." "The Elder Scrolls." "Doom." "Fallout." And "Warcraft." Beyond that, it gives them huge amounts of development talent that understands real, and virtual worlds. Plus the technologies that understand how to manage millions of real-time users in a three-dimensional environment. Every other major technology, and entertainment brand from Disney to Apple to Sony is right in the middle of working out their metaverse strategy. So how do you fit in? Now, companies are getting ready to provide you with the tools that are going to fundamentally change how you do business and leisure. Think about how the internet changed everything. It's about to happen again in the metaverse. So get ready for a new world coming sooner than you think.

Development Tools

No-code

(smooth music) - Remember a few years ago there was a massive push to get more people interested in tech where we were told that everyone should learn how to code? Now we hear that no-code or low-code is the way forward. What is going on here? Hey, there. I'm Walt Ritscher. I'm a programmer and I enjoy writing code, but I think that this no-code movement is real, is poised to grow, and it's worth exploring. It's hard to imagine how our world would look without software. We've successfully combined code with hardware devices of all kinds. Software permeates everything. It's in your computer, your laptop, and phone. But look around you. It's in your TV. It powers airplanes and space flight. Out on the farm, it's embedded in the wheat harvester. It's a key part of modern medical equipment. It really affects everything we touch. Systems like these are built with traditional programming languages by professional programming teams. And it's the same at work. Most company applications are built by the professionals, and yet there are millions of potential application ideas that never get built. Don't believe me? Ask any department at your company about it. You'll hear stories about missing features in existing applications. Ask them about what would make their work easier, and you'll hear wonderful ideas for small applications and tools. Let's put this in perspective. There are 27 million programmers worldwide, with about 14 million professionals. We simply don't have enough of them to build everything needed at your company. Compare that to the billions of people who use office suites like Google Workplace and Office 365. They are already adopting these suites to fill the gaps. How much of your company is run from departmental spreadsheets? Give people better tools and they will make better apps. Before going further, let's define no-code systems. Most of them are visual drag-and-drop tool sets. With these tools, people can connect predefined components to build a mobile or web application. Users can create an application interface by moving UI parts around the screen. Users can piece together the application UI and connect it to data as if they were building a flow chart. They can write customized logic as if they were writing Excel functions. Now, professional developers can use these systems to quickly build applications without writing much code, but there is another audience adopting these tools. Yes, I'm talking about the millions of power users out there, the business analysts, office administrators, and small business owners. It's the domain experts in your company who understand the business needs and rules. They have great ideas and good technical skills, but little knowledge of traditional programming languages and no desire to learn. Microsoft coined a phrase for this demographic, the citizen developer. Here's how Gartner defines that term. A citizen developer is an employee who creates application capabilities for consumption by themselves or others using tools that are not actively forbidden by IT or business units. What are those tools? What does Gartner mean? There are many companies entering this market. Microsoft's Power Platform is a popular choice and it's the one I'm most familiar with, so we'll look at it for a few minutes. Be assured there are comparable and interesting tools made by other companies. Power Platform consists of several parts. Power BI, a business intelligence tool for analyzing and visualizing data, Power Automate, a workflow and automation tool, Power Virtual Agents, a low-code way to make chatbots. To round out this list, the centerpiece of Power Platforms is Power Apps. It is a service for building custom business apps. It comes pre-built with standardized data connectors and services for working with your company assets, whether they are in a SharePoint table, an Excel spreadsheet, or official company database. When ready, the Power App can be instantly shared to your team or the entire company. No need to understand how mobile app stores work because the app is automatically available on the web and mobile devices. Are you skeptical? If so, here's my story. I created a data entry prototype for my team using data in SharePoint. Total time to create and deploy as a mobile application was 20 minutes. I believe we will see a rapid rise in adoption of low-code systems. The Power Platform tools are only a click away on the Office 365 website. That makes them easy to find, which means people discover them. Microsoft's statistics show that people are embracing the tools and creating their own applications. It's the same for the other no-code platforms. Gartner estimates that the low-code market will grow to $30 billion by 2025. They also forecast that by 2024, low-code application development will account for 65% of all application development activity, mostly for small and medium-sized projects. Here's my final thoughts. The no-code platforms are available now and they enable every person in your organization to build applications within the company and share them with their team members. They are easily deployed to mobile devices or the web. They are built on officially sanctioned platforms, so the concern about shadow IT is low. This is the promise. Whether you are a software developer or not, if you can dream it, you can create it.

Cross-platform development tools

(smooth music) - Let's say you're planning a new software product. Your app should run on Android, but it also has to work on iPhones and iPads. And maybe you also want to create desktop versions for Windows and Mac, and even a progressive web application that runs in a browser. If that describes your development plans, you might want to look at cross-platform development toolkits. These are tools that let you build some or all of an app in a single code base and deploy on multiple operating systems and platforms. This sounds like a simple choice. Create many native versions, one for each platform, or build it just once. Separate native apps can give you great performance and each app can be designed specifically for its target platform. But there's an obvious cost. You'll need developers who know each of these platforms' supported languages, Swift for iOS and Mac OS, Kotlin for Android, C# for Windows, and JavaScript for the web. And in addition to the initial development cost, you'll have to maintain all of those versions for the software's lifetime. So, that takes you to cross-platform development tools. Development teams choose these tools for one reason, to use their resources efficiently. Why build the same app four or five times when you can create a single code base? And there are many cross-platform toolkits available. There are potential hazards. The history of these tools is littered with projects that showed initial promise, but fizzled out. The Titanium SDK from Appcelerator and PhoneGap, the commercial version of Apache Cordova, both supported programming with Javascript and packaging for Android and iOS. Both products have been retired by their original sponsors and are now being made available as open source projects so that existing apps can be maintained. But neither of those older frameworks is a good choice for new apps. So, which cross-platform tools are available today that have both promise for the future and ongoing support? Ionic and React Native are current toolkits that let you build apps with web development technologies, such as HTML and CSS. Both platforms use JavaScript and both offer pre-built UI components, but there are differences. Ionic is backed by an open source organization, while React Native was created by and is supported by Facebook. React Native apps are primarily designed for deployment on iOS and Android, while Ionic also supports Windows. In both cases, there are third-party tools for deploying these applications on other platforms. If you're looking for a toolkit that leverages the existing skills of web developers, both React Native and Ionic deserve a look. On the other hand, apps built with web technologies don't always look and behave like natively-built apps, and they can't compete on performance. If you want your app to look like a native app and to be really fast, you might look at two other toolkits. Xamarin is a cross-platform toolkit that's owned and supported by Microsoft. With Xamarin, you code in C# with the .NET framework. Xamarin.Forms is a subset of this project that provides the user interface. Apps built with Xamarin can be distributed on Android, iPhone, iPad, Mac, and of course, Windows. And because it's a Microsoft project, developers get tight integration with Visual Studio, Microsoft's premier development environment. And then there's Flutter, a cross-platform toolkit that's supported by Google. Flutter apps are built with Dart, a relatively new language compared with, say, JavaScript or C#. The resulting apps can be distributed on Android, iOS, Windows, Mac, the web, and even embedded environments. Most importantly, Flutter-based apps are super fast. Their performance can exceed that of apps built with native SDKs for iOS and Android. Cross-platform development tools have come and gone, and investing in one of those toolkits takes research, experimentation, and a certain level of trust in the vendors who create and support them. But if you can find the right toolkit for your project, you can spare yourself the cost of creating and maintaining completely different versions of an app that has to run on many platforms.

Excel and JavaScript

(upbeat music) - What do the world's most popular spreadsheet and one of the most popular programming languages have in common? Until November of 2021, not much. But then Microsoft made an announcement that had longtime users of the company's Excel spreadsheet product do a comedic double-take. Excel would now support a set of JavaScript based APIs that would let developers create their own custom data types and functions. What exactly does this mean? What does JavaScript, a language most commonly associated with programming webpages have to do with the business oriented functionality you associate with Excel? Well first, the new JavaScript APIs for Office add ins don't have anything to do with web pages. While JavaScript is commonly associated with front end web development, it's just a programming language, but it's one of the world's most widely used programming languages. So by opening up Excel to JavaScript, Microsoft leverages a massive community of developers who can take advantage of these new APIs. Now this new feature set could be confused with another tool that Microsoft released in 2020. If you have a commercial or educational Office 365 license, you may have discovered a new automate menu choice in the web based version of Excel. It opens an interface that lets you record and playback actions such as entering data and formulas or formatting output. These recordings are saved in a language called TypeScript, a super set of JavaScript, and you can customize those scripts or build new ones from scratch. This is a great feature, but it's only available to certain customers and it only works in web based Excel, not in the more powerful desktop releases for Windows or Mac OS. The new JavaScript APIs that Microsoft announced at the 2021 Ignite Conference are designed for use in what are known as Office add ins, libraries that any developer can create and distribute and that work on most platforms where you can use Microsoft Excel. Unlike the Office Scripts feature, Office add ins work for anyone who has those software products, not just those with particular licenses or on a single platform. There are two new JavaScript APIs. The Excel JavaScript API lets you use strongly typed software objects to access worksheets, ranges, tables, charts, and a lot more. And the common API gives you access to features such as the user interface, dialogues and client settings. There's extensive documentation for these new APIs, including plenty of sample scripts that you can copy and paste into your own projects. These new tools also let developers create their own custom data types that manage images, arrays, and more. Prior to the introduction of custom data types, the API had supported simple values, such as numbers and strings. Developers can now define data types that express the formatting of those values. So something as simple as converting a raw number to a currency formatted value can be defined once and then used multiple times. And the developer can also define custom functions. As of late 2021, this feature was available as a preview only on Office for Windows and not across platforms. But if you're a JavaScript developer and you have a Windows based development system, you can start experimenting with this capability right now. JavaScript and Excel may not seem like the most likely of technology partners, but by bringing these two tech communities together, Microsoft may be realizing the old maxim that the whole can be greater than the sum of its parts, especially when you calculate that sum with a spreadsheet and an incredibly popular programming language.

Security
Zero trust
(futuristic upbeat music) - For a long time, the security model used by corporate networks has been based on the idea of a security perimeter that is protected resources are connected to a network that's kept secure from the outside and any device within that protected network is able to use resources available on the network. And that model worked pretty well for a long time but times have changed. We don't all work in an office on a PC wired to the wall using internal systems anymore. Now we have to consider things like cloud services, mobile devices, remote workers threats from bad actors, ransomware, data breaches, and so many other things that the old model of a single secure perimeter just isn't well equipped to handle. In fact, many companies don't even have a single physical location where work takes place and all their resources are accessed remotely by geographically diverse employees. So how do we keep up with these changes and access requirements while maintaining security? We get rid of that secure perimeter and move security evaluation to individual services, devices and users. This is called the zero trust security model. And while the ideas behind it have been around for decades, it's starting to gain traction in our evermore connected world. Zero trust turns our legacy model of network security almost inside out. Instead of maintaining a secure perimeter within which everything is automatically trusted, with the zero trust, there is no single perimeter and each device and interaction between systems comes with its own authentication and authorization steps rather than being trusted simply because it's within a secure network. In this model, individual to devices, users, services and interactions are specifically authorized using a variety of different systems. Devices like computers, tablets and phones are enrolled into an access management system and are provided certificates or other credentials they use to prove their identity whether they're attached to the corporate network through ethernet or they're on a wireless connection at a cafe, the access management system determines whether devices are what they say they are and whether they're allowed access to protected resources. These access management systems can also evaluate device health. That is whether a device is running a specific version of software, whether it has unapproved software installed and so on to protect a system from unauthorized or compromised devices. If a device check fails or if a user tries to access protected resources from an unapproved device, their request can be denied. User control is handled by identity providers, services which designate which users are allowed to use which services and which devices for which tasks. Users might use a username and password and they may also have a certificate or other token that provides a second security factor. In many cases, a user's identity is treated as a single entity, that is we don't have one username and password for one service and a different set of credentials for another. Our identity token is used for services we access, ensuring that access is granted to a person not simply to a username. Resources that users need to access also vouch for themselves using similar services. When we access a resource like a database or a payroll system, our device and the security system check that the remote resource is what it says it is to avoid impersonation of secure resources. This is called mutual authentication and it's a core principle of the zero trust model. Another core principle of zero trust is that there's no implicit trust of a device, of a user or of a resource. It doesn't matter if the CEO's computer is plugged into the corporate network in her office at the company headquarters. It's still subject to all these control checks at various levels for each task it needs to perform. Imagine that you need to access some information from a database. In a zero trust system, when you make that request, the system will check whether the device you're using is approved and it will check whether you're allowed access

to it. The system will check whether you're permitted to access the specific data you're requesting. You might be provided a session key that is an authorization specific to the single interaction you're performing which can't be used for any other request. Each of these checks helps to control the security of information and helps to prevent bad actors or malware from spreading throughout a network. And in most cases, each of these verifications and requests will be logged in an auditable system. Intelligent systems can be used at many points of this process as well to determine suspicious patterns of access or to learn what likely patterns of access look like. This can help administrators identify malicious activity more easily than with manual methods. All these interactions take place through encrypted communication channels, making it more difficult for bad actors to read the data exchange between devices. This approach of using many layers of security is often called defense in depth. Zero trust is becoming a key concept in designing information access systems and cloud and enterprise service providers like Microsoft, Google, Amazon, Oracle and many others offer case studies, white papers, reports and other resources to demonstrate its benefits and how it can be used in a variety of scenarios. Building a zero trust system takes planning and coordination and how you might apply it will depend on your needs and security requirements. The amount of access controls and levels of security that enables zero trust can sound a little bit paranoid but this model enables us to securely resources in the cloud or in our data center from mobile devices or from onsite systems while allowing specific controls and auditable logs. We'll hear a lot more about zero trust in coming years as the model is more widely adopted. And while we can't fully prevent security problems, the zero trust security model provides a robust way of mitigating security risks while enabling the modern mobile workforce.

Technology Business

Why is there a chip shortage?

(gentle music) - Microchips, usually called chips for short, power the modern economy, and can be found in everything from computers to cars, from tablets to toasters, from cool toys to cooking tools. But if they're so widely used, why is there suddenly a chip shortage? Well, the easy answer to that is because of everything. But instead of taking the easy answer, let's dig into the real reasons behind the chip shortage. There isn't just one problem, or bottleneck, creating this constrained supply. In the aftermath of the COVID-19 pandemic, global shipping became unbalanced, and now many products, including chips, aren't able to get from manufacturers to their customers. In ports around the world, container ships sit waiting to deliver their cargo, and to pick up new cargo to travel back across the sea. And with passenger flights at lower levels than before, air cargo capacity is also constrained. So it's harder for things to move around. And this compounds the shortage of many products, which have components made in one place, which are assembled into larger systems in another place, which are then used to build a final product somewhere else. This is a problem for manufacturers in places like the United States, because most of the world's highest capacity and most modern chip factories, which are called fabrication plants or fabs, are located in Europe, the Middle East, and Southeast Asia. This is changing, and there's an effort to open new chip fabs in the U.S. and elsewhere, to help increase geographic diversity of manufacturing. But fabs use some of the most complex technology and processes of anything, that we as humans are capable of producing. So getting these factories up and running will take some time, probably at least a few years. And by the time those are built, there's the risk that technology will have improved even further in the meantime. Another interesting factor contributing to the shortage of chips for some industries, is the timing and size of orders placed with fabs by customers of various kinds. Early in the global pandemic, many car manufacturers canceled orders for chips in response to predicted low demand for new cars. Though, as things started to normalize again and demand for cars rose, those manufacturers had lost their place in line and had to place orders after other competitors, inside, and outside, their industry. Because fabs have a finite capacity to produce a given number of chips in a given period of time. There's an upper limit to the number of chips that can be made in a particular month, or a quarter, or year, and everyone else has to wait their turn. Related to this, there were rumors that many organizations who use chips in various products, have been over ordering to make sure they can get enough. And that hoarding has further restricted the supply of chips and chip production for other purposes. And another reason contributing to the shortage that we can't overlook, is that, as in many other industries, illnesses, closures, and the resulting capacity reductions at the factories that make chips, also affected and continue to the effect, the number of chips produced. Delays in shipping, also affect the availability of raw materials that fabs need. Chips are made from silicon, and silicon is refined from silicate materials, like sand and other minerals, which are often mined far away from the industrial centers, where fabs turn purified silicon wafers into chips. The same safety closures and shutdowns that affect factories also affect mining and refining operations. All of these causes combine to make a shortage of chips that itself sends ripples throughout the supply chain of many other products. Many of the chips we think about in the technological world are advanced, powerful marvels of engineering, running VR headsets and video game consoles, or doing computations in cutting-edge supercomputers. And shortages of these new model chips map directly to delays in getting that latest laptop, or smartwatch, or fancy TV. But chips appear in many places, often doing unglamorous things, like controlling the valves in the car's engine, keeping time in alarm clocks, or opening the garage door. Many of the chips that perform these tasks were designed years, even decades ago, and continue to be made according to the same plans. But even these older model chips still need to be manufactured now. For practical reasons, and because many industries have shifted to just-in-time manufacturing, there isn't a giant stockpile of these chips that manufacturers can turn to instead. And in many cases, simply switching an existing design to use a slightly different, or more readily available chip, will be met with increased costs, for updating plans and tools, for testing, and for changes to specifications. In the case of component parts of cars and other products with lots of safety regulations, new designs incur further delays for regulatory approval. So until the supply chain gets sorted out, one way or another, we'll continue to see delays in production of all kinds of products, large and small, expensive and inexpensive. And that means we will need to wait to buy that new laptop or that new car.

Windows Store Improvements

(upbeat music) - Hi, I'm Walt Ritscher. Windows 11 will release in 2021. It has improvements in many areas, some are consumer focused, like the new user interface and icon sets. I'm a Windows programmer. I'm interested in what's new for software developers. There's plenty to discover, like Windows Widgets, there is a new UI toolkit available, which makes your apps modern looking and there are tools for cross-platform development. So you can have one code base for apps that work on multiple operating systems. You may have heard there are changes coming in the Microsoft Store. It's true. The updated Microsoft Store supports an extensive range of application types, which makes it easier for you to deploy to Windows, reach a wider audience and make more money too. Let's look at some of the improvements. Here is a blog post about the Microsoft Store updates. The post has information for consumers and programmers. Let's scroll down to this section, introducing a new Microsoft Store. To be clear, Windows 11 is mostly an incremental update to Windows 10, therefore many of the improvements will affect both versions and that is true about the Store as well. Now we'll scroll down the page, there's a section on consumer features, including one about a pop-up store that helps install apps from the web. The section we're interested in as programmers is here, a more open Microsoft Store for developers, and we'll start talking about support for more types of apps. The first thing to note is there are very few restrictions on what type of applications can be published in the Store. Windows developers can publish their application in the Store, no matter what framework they are built upon or what packaging technology is used. This is great news for developers as there is now a single marketplace for applications available on Windows. Gone are the days where the Store only contained universal Windows platform apps. Now you can publish WinForms or Windows Presentation Foundation or C++ win32 desktop apps. Also the Store supports Electron, React Native, Xamarin and Java apps. And for web developers, you can publish your progressive web apps, also known as PWAs in the Store. Much of this new feature is enabled by the changes to the application installers. This initiative started in Windows 10 with the addition of a new installer packaging service called MSIX. As you can see here in the Microsoft Docs, there are sections on how to publish to the Store based on your installer type. So here's the one for publishing an MSIX app to the Microsoft Store. Here's information on app submissions, how to set pricing and availability, how to set age ratings, how to upload your packages, and then much more. And there's sections for each type of app. So there's one here for executable and the older MSI installer based applications and another one for how to publish your PWA app to the Microsoft Store. Once your app is in the Microsoft Store, it is available to over one billion consumers. That's a big audience. Now what about selling your app? Microsoft announced it is changing its Revenue Sharing policy. The new Store will let vendors use their own commerce engines inside the interface. In other words, if you've created a way of upselling to a pro version or provide subscription services or micro-transactions, then you get to keep all the revenue. On the other hand, if you don't want to build a commerce system, you can choose to use the Microsoft Commerce Service and you'll pay the 15% revenue share to them for apps, or 12% for games. Microsoft's Commerce System is robust. I've used it on some of my apps. It's simple, has a global audience, and you can pick regions where your app is available in the Store, and it supports many types of revenue streams. That's a quick look at the changes for developers in the Windows 11 Microsoft Store. We can publish nearly any type of application, reach a large global audience and keep 100% of our revenue when we have our own commerce engine. Microsoft says that they intend to publish all their flagship desktop products this way. They've announced a few like Visual Studio and Microsoft Teams. Another surprising announcement about this release is that Windows 11 will let users run Android apps on Windows based computers, and there'll be able to discover and install those apps through the Microsoft Store. This is encouraging news for Android developers. You can get more details about how Microsoft is enabling Android on Windows elsewhere in this course.

Android apps on Windows 11

(upbeat music) - I'm David Gassner, and this is another installment of Tech Trends. On June 24th, 2021, Microsoft made an announcement that could be a game changer for Android developers. Windows 11 will let users run Android apps on Windows based computers in much the same way that Google's Chromo OS does on Chromebooks. And they'll be able to discover and install these apps through a redesigned Windows store. The first question many were asking after picking themselves up off the floor was, how will this work? On Windows 8 and Windows 10, the Windows app store was only able to distribute what Microsoft calls universal apps. With Windows 11, developers will be able to distribute both universal and conventional desktop applications through the store, but also apps that were originally built for Android phones and tablets. The Windows Store will delegate distribution of Android apps to the Amazon Appstore. The first time user tries to install an Android app, they'll be asked for us to install the Amazon Appstore, and then they'll be able to install other apps. The Amazon Appstore will process app purchases and provide important runtime services. But how will these apps that were designed for Android devices be able to run on an Intel or an AMD based computer. To understand this, let's review how Android apps run in normal conditions. When you package an Android app for deployment, the code you write in Java or Kotlin is compiled into something called DEX byte code. On each Android device, that byte code is recompiled by the Android runtime, or ART, into something that works on that specific device. This process is known as ahead of time compilation, and it's what makes it possible for Android apps to run on diverse devices with processors from Qualcomm, Samsung, and other chip makers. On Windows 11 computers and tablets, something similar will happen. A technology called the Intel Bridge will recompile the app to work on a computer with an Intel or AMD processor. In theory, this means that your app built with Java or Kotlin will just work on Windows 11. Will this turn out to be true in practice? We'll have to wait and see. It's truly early days. So if you're an Android developer, what should you do now? If you haven't already, you may want to explore getting your app into the Amazon Appstore. Many developers haven't done this for a variety of reasons, but it really comes down to this. Until now, the Amazon Appstore has mainly been used to distribute apps on Amazon Fire tablets and the number of those devices in user's hands is dwarfed by the population of cell phones and tablets that support the Google Play Store. Many developers felt the extra work just wouldn't be worth it. But with Windows 11, that calculation changes. There are billions of Windows users, and each one who upgrades to Windows 11 becomes a potential customer for your Android app. There may be some work involved. If your app uses functionality that depends on Google Play services, such as in-app purchases or the Google Maps API, you may have to rebuild those features with Amazon's equivalent tools. You'll want to learn how to use build variants, which let you create alternative app IDs, multiple versions of resources, such as image files, and different code paths that are used depending on which variant is installed. Also, make sure your app looks great and behaves well on larger screens. The Android SDK has all the tools to make this happen, but it isn't automatic. You have to create alternate layouts for different screens and manage them with fragments. Your apps should adapt gracefully to a variety of screen sizes and pixel densities, so it works well on cell phones, tablets, and full-size computer displays. If you're an Android developer and you want to distribute your Android app in this new and potentially massive marketplace, go create an Amazon Developer account, it's free, and you'll be able to explore the tools that Amazon provides as alternatives to Google Play's APIs. And when this new capability comes to Windows 11 users, your Android app can be ready.

Chapter 7

Why focus on emerging technologies?

- We are today in the middle of a technology revolution more disruptive than any other in human history. - Figuring out how to start a fire on demand and the invention of the wheel were indeed profound developments. However, these developments took centuries, if not millennia, to diffuse globally and change how people lived and worked. - Today's digital technologies are emerging at a far greater speed and with greater year-to-year impact than even other development of recent decades, such as the PC, the Internet, and the mobile phone. - Artificial intelligence is clearly the most important emerging, digital technology today. Focusing on its promise, Google CEO termed it "more profound than electricity or fire." Looking at the risks, however, the late physicist Stephen Hawking worried that AI could spell the end of the human race. Microsoft's Bill Gates has put himself in the camp of the alarmists. - AI is not the only driver of today's digital disruption. Others include Internet of things, robotics, and blockchain. No industry, no company, and no individual can escape the influence of these developments. - Hi, I'm Anil Gupta, Michael Dingman Chair and Professor of Strategy and Globalization at the Smith School of Business, the University of Maryland at College Park. - And I'm Haiyan Wang, Managing Partner of the China India Institute. To set your organization apart, it is imperative to understand the key features of emerging digital technologies and how some organizations are harnessing their power to benefit society. - This is a companion to our course on Digital Strategy. Here, we will examine the practical application of these technologies in various industries. - We invite you to join us on this journey.

Artificial Intelligence

A brief primer on AI

- Artificial intelligence or AI refers to software algorithms that attempt to replicate tasks that have historically required human intelligence. AI applications are now becoming embedded in almost all aspects of economic, social, and personal life, managing mutual funds, helping you find a better romantic partner, or even personalized counseling. Early efforts at developing AI focus largely on feeding highly codified knowledge to computers and then trying to have them mimic human decision-making. For example, researchers would interview doctors, try to codify their decision-making rules, and then develop expert systems embedded with the codified knowledge. This approach was not only time-consuming but also started reaching a plateau because it became evident that human decision-making rules could be codified to only a limited extent. Over the last decade however, computer scientists have been tackling the AI problem in a fundamentally different and far more promising way. How do toddlers learn to walk? Certainly not by first learning physics and biology. Rather they learn via trial and error. They observe adults walking, eager to mimic. They try something and fall down. They see what works and what doesn't. Learning from this ground-level data, they try again, perhaps a bit differently. Eventually, they figure it out. Could machines be programmed to learn similarly? This type of machine learning is the core paradigm in AI today. Given enough data, when machines learn on their own they can get smarter very rapidly. Importantly, they can even learn to make good decisions in areas where humans may not have any codified knowledge at all. Computer vision and natural language processing are two of the biggest areas where AI technologies are being applied today.

Case study: Diagnosing lung cancer (Google)
- Lung cancer has the highest rate of mortality among all cancers. Accounting for an estimated 160,000 deaths in the United States in 2018. Early detection has been shown to reduce mortality by 20 to 43%. That's why medical guidelines endorse the use of low dose CT imaging for lung cancer screening. Despite the guidelines, even in a rich economy such as the United States, the healthcare system suffers from challenges. Radiologists are prone to erroneous conclusions. And there is significant variability in the skill levels and findings across radiologists. Emerging economies suffer from additional problems. The shortage of trained radiologists far exceeds the shortage of imaging centers. In May 2019 scientists at Northwestern University and Google co-authored a paper in "Nature Medicine" detailing the results of tests with an artificial intelligence system trained to read low dose CT images. The scientists also compared the accuracy of the AI systems diagnoses with those of six expert radiologists. This is a leading example of the application of AI to computer vision. In cases where they could look at both the recent scan and an earlier one the AI system performed as well as the radiologists. However, if both looked at only the more recent scan the AI system beat the radiologists. The AI system also produced fewer false positives, that is incorrectly concluding the presence of cancer. As well was fewer false negatives. That is incorrectly concluding its absence. Thereby reducing unnecessary follow up procedures and cases of missed tumor. Additionally, the system was able to pinpoint regions of the lung which carried a higher risk of malignancy. So, what were the sources of the AI system's advantages? First, the system was not bound by the constraints of the human eye. Which can look at only 2D image slices on the screen. Instead, the system could look at the entire 3D image in one go. In fact, because the system could look at both the recent and the earlier 3D images simultaneously, it really had 4D vision with time as the fourth dimension. Second, because of it's far greater computational power the system could analyze each pixel as well as various combinations of pixels with far greater granularity than human radiologists could. Third, the system relied on deep learning. A more recent branch of AI. In deep learning, the AI is trained on a subset of cases where it is given anonymized scans and is told only whether or not later tests would reveal the presence or absence of cancer. With no prior medical expertise coded in the system had to learn on its own how to make better predictions about the presence or absence of cancer through a rapid fire sequence of trials and errors. This is a process very much akin to how a child learns how to crawl, stand up, walk, and run. This blank slate approach made the system open to the discovery of novel signals of cancer in the CT scans. As a result, the AI system can now begin to teach human radiologists, an interesting case of a reverse learning. The utility of these type of system should be even greater in emerging economies as it can help fill in the gap caused by the scarcity of trained radiologists.

Case study: Crunching emerging market knowledge for investors (Arkera)
- The rise of emerging markets has made it essential for investors on Wall Street and other financial centers to develop a deep understanding of the opportunities and risks across the entire world, not just in the U.S. and Europe and Japan. How likely are lawmakers in Brazil to pass a controversial bill to overhaul the pension system? What are the odds of India's ruling party winning upcoming local elections in the biggest state? Should we bet on the Nigerian president liberalizing the agricultural sector? Trying to answer these questions by reading online English versions of each country's main newspapers could never give any investor any type of information or analysis advantage simply because doing so merely scratches the surface of available and relevant information. So what could the savvy investor

do? The answer lies in using the power of AI to browse through, digest and synthesize millions of bits of information, not just in newspapers but also on TV broadcasts, Web pages and social media feeds such as Twitter, much of it in local languages. Startups such as Arkera in London and Sigmoid in New York do exactly this and have found ready clients around the world. In the case of the Brazilian pension overhaul bill, Arkera's AI system kept a close watch on everyone of the 500 certain legislators. A few weeks before the bill's passage in July 2019, the AI system was able to predict the outcome almost to the letter, giving its hedge-fund clients significant early mover advantage regarding movements in Brazil's currency. So how does the AI system do this? The starting point is that computers can access far more data points, including social media feeds, simultaneously than humans ever could. Further, AI driven translation systems are now able to translate all of these into English with almost the same degree of accuracy as that of human translators. The biggest value of AI algorithms, however, lies in making sense of what all of this information means. Some of the early work in this area began with scientists training algorithms to understand of content of first sentences and then paragraphs. For example, the AI would be given thousands of paragraphs as well as questions such as, "Does John like Jane?", and the right and well as wrong answers. Through a process of trial and error learning, the AI would eventually become smart enough to understand the rules of grammar and make sense of new previously unseen paragraphs. Over time, scientists in the field of natural language processing have been able to develop AI systems that can understand movie scripts, summarize CEOs' earnings calls, and even write drafts of newspaper articles by first digesting what's being said around the world on a particular issue. The leap from these capabilities to the AI systems of Arkera and Sigmoid was not particularly dramatic. Given the perpetual arms race among those who develop these technologies as well as those who use them, it's a safe bet that natural language processing will become far smarter and far more widely used within just the next three to five years.

Internet of Things

A brief primer on Internet of Things

- The internet of things, IoT, refers to a system of objects, as well as even people and animals, who carry unique identifiers and can transfer data over the internet without any human-to-human or human-to-computer interaction. We can think of the IoT as an internet-based network, where the connected things are nodes of the network. A sensor embedded in an aircraft engine is an IoT node. So is an internet-connected doorbell, or a camera, a smartphone, or a driverless car. Even a bird, a cow, or a whale, when tethered with an internet-connected sensor, becomes a node in the IoT network. This would be true also of any person who gets a pacemaker or a chip implanted in his or her body, which can then transfer data over the internet. IoT sensors are spreading like wildfire, and have already been put to use in a wide range of applications, from homes to hospitals, to factories, to farm machinery, and even entire cities. According to some estimates, by the end of 2020, there would be over 10 billion IoT devices worldwide.

Case study: How the oil and gas industry has embraced IoT

- Oil and gas is one of the largest industries in the world. Accounting for revenues exceeding two trillion dollars. Given some of the industry's unique characteristics oil and gas companies have been among the earliest and most enthusiastic adopters of IoT technology. Let's see why. First, oil, gas, and petrochemicals are commodities. Every company, no matter how large is at the mercy of market prices. Thus, operational efficiency is the primary determinant of profit margins. Second, unlike other process industries such as steel or paper, accidents in oil and gas can be not only deadly but extremely expensive. In terms of cleanup operations and lost production. Third, many segments of the industry, for example oil drilling and refining, can have significant and very costly environmental consequences thereby requiring extreme preventive action. Here are some of the ways in which oil and gas companies use IoT technology to address the imperatives of operational efficiency, accident prevention, and environmental sustainability. A large oil and gas company can have over 50,000 wells and multiple pieces of equipment at each well. The failure of a single machine at a well can lead to lost production while also falling foul of regulations. Internet connected sensors can give early warning of impending failures or immediate alerts in case of actual failures. Something that would be impossible to do with in person field inspections. Aggregated over all wells, data from sensors can also help managers learn why some wells perform better than others so that the best practices can be diffused company wide resulting in improved efficiency. In refining operations, IoT sensors can monitor emissions, waste, and environmental conditions on a 24 by seven basis. This approach not only reduces the cost of onsite inspections but is much more thorough. It also enables the company to be highly responsive thereby reducing penalties in the event of a mishap. IoT sensors embedded in pipelines, trucks, tanks, and in the case of home heating oil even customer's homes, can enable the company to manage its entire supply chain far more efficiently and with greater responsiveness to fluctuations in market demand. Industry insiders estimate that IoT implementation is already adding several billion dollars in economic value to the oil and gas industry. The global demand for IoT sensors and related software and services for this industry is predicted to exceed 30 billion dollars in 2025. Similar implementations of IoT are underway in other industries from steel mills, to forestry and paper, to auto factories. Ongoing developments such as cheaper, smaller, and more powerful sensors that are connected to the internet where 5G and can leverage the growing power of data analytics in AI seem set to make IoT as ubiquitous as the clothes we wear or the air we breathe.

Cases study: How IoT is transforming healthcare

- Most countries, rich or poor, are groaning under the weight of rising healthcare costs. In rich countries, these costs account for 12% of GDP, with US as an extreme outlier at 18%. Not surprisingly hospitals as well as individuals are becoming enthusiastic adopters of the IoT technology. It helps improve quality of care without much, if any, increase in costs. Let's look at some key examples. Smart phone like devices can now take photos of wounds and transmit them to the patient's medical file. These images help nurses assess the severity of a wound and track how it's healing over time. Smart pacemakers now come with the ability to communicate with external monitors. When a patient with such a pacemaker implanted in his or her heart visits the doctor, the doctor can download information about the behavior of the heart as well as whether or not the device has sufficient battery power. A growing number of IV machines are embedded with sensors that can be programmed to deliver the right amount of medication at the right time. In the case of pain medication the patient could also be given a wristband which can communicate wirelessly with the IV machine. When the patient needs to reduce the pain he or she can use the wristband to trigger the machine to deliver medication. The system is designed to prevent overdosing. Mount Sinai Hospital in New York City has installed sensors in hospital beds that let the staff know exactly when a bed has become free, is ready for reuse, and its location. This simple move has reduced wait times by several hours for half of the hospitals emergency patients. Imaging machines such as for taking x-rays, CT scans, and MRI images routinely come embedded with sensors. These play several roles. They monitor the performance of the machine in terms of how well it did the task it was designed to do. They issue alerts if during the machines use certain thresholds get exceeded. And they trigger calls for preventive maintenance so that unwarranted downtime is reduced or eliminated. As sensors become more powerful and cheaper they're bound to become even more pervasive throughout the healthcare system.

Robotics

A brief primer on robotics

- Robotics refers to the technology behind robots. That is, machines that are programmable by a computer and which can carry out a complex series of actions automatically. Technically, the term robot can be applied to physical machines such as a drone or purely digital ones such as a voice assistant on a smart phone. In common usage, however, we associate the term robot with physical machines, and the shorter term, bot, with purely digital ones. In this course, we will focus only on physical robots. Robots are evolving rapidly in terms of both their physical dexterity as well as intelligence. This evolution is being propelled by a combination of developments in both hardware technology, as well as artificial intelligence. Robots can vary widely in terms of the degree of their autonomy. Some of the earliest robots in factories, such as an automotive assembly plant, were not capable for much, if any, autonomy. They were programmed to perform a series of tasks in a well-defined pattern. Without reprogramming, they could not deviate from this pattern. In contrast, a car running on autopilot mode today, is capable for a high degree of autonomy, at least on highways. Scientists and engineers are pushing robotics technology along a number of important dimensions. They are improving the physical dexterity of robots so they can mimic the human hand, running, jumping, climbing stairs, and so forth. They are making them much smaller so that they can go inside the human body to perform surgery or act as an army of artificial insects during warfare. They are experimenting with soft robots that are powered by chemical reactions in the robot's brain, thereby obviating the need for electricity. Most important of all, robots are becoming more intelligent and thus more autonomous.

Case study: Robot-assisted surgery (da Vinci surgical systems)

- We believe that the da Vinci system is a forerunner of how a majority of complex surgeries will be done over the coming decade or two, at least in the developed economies. With aging populations, the market size will grow faster than the population. As new entrants go after this opportunity prices are bound to drop significantly. The da Vinci Surgical System is the world's most advanced robotic system for minimally-invasive surgeries. The da Vinci system is particularly well-suited for surgeries that require a very high degree of precision, dexterity, and control. In a sense, this robotic system expands the surgeon's capabilities. There are benefits for the patient too who undergoes only a one to two-centimeter-long incision resulting in much less trauma, much faster healing, and shorter hospital stays as compared with open surgeries. These benefits come with very high upfront investment by the hospital. Each da Vinci system costs over two million dollars. There are also sizeable annual maintenance fees. It also requires significant training for the surgeon in terms of how to use the system. As a result, such systems are economically viable primarily for those centers that perform a large number of surgeries each year. The system is used for a wide range of surgeries including cardiac, urologic, gynecologic, pediatric, as well as general surgery. Estimates suggest that three-quarters of all prostate cancer surgeries in the United States currently use da Vinci robots. Let's look at some details of how the da Vinci system works. The most popular model, da Vinci SI, consists of two separate but connected sections. One section is a robotic tower with four arms. Three of these hold various surgical instruments. The fourth one holds 3D high-definition cameras. The second section is a console where the surgeon sits and operates the robot's arms while looking through two stereoscopic monitors. Additional video screens enable other members of the surgical team to observe what the surgeon sees. In manipulating the robot's arms, the console permits the surgeon to exercise fingertip-level precision over the surgical instruments.

Case study: Robotic assistants in supermarkets (Walmart)
- Within a decade after Sam Walton founded Walmart in 1962, the company started acquiring a well deserved reputation as the most responsive and most efficient retailer in the United States. Walmart's larger and more tailored product mix, as well as lower prices, were the most direct cause behind the decimation of old stalwarts, such as Sears, Montgomery Ward, and K-Mart. Over the last two decades however, Amazon has proved to be not just Walmart's match, but by doing away with the cost of physical stores altogether, even more efficient. After initial struggles in figuring out how to fight back, Walmart has now emerged as perhaps the most nimble traditional retailer in the world. There are two major parts to Walmart's strategy for competing with Amazon. After acquiring Jet.com in the US, Flipcart in India, as well as other Etailers, it is going head to head against Amazon in online commerce. Equally importantly, Walmart is turning to technology to make its physical stores even more efficient. An army of robots is playing an important role in this effort. Having learned from several well-designed experiments, Walmart is now in a process of deploying thousands of robots in a large number of its over 11 thousand stores. Some of these robots perform backroom operations such as unloading trucks and then scanning boxes. This is where Walmart made its early moves in bringing robots to the stores. Others are pickup towers, where consumers can pick up goods ordered online. More recently, however, robots have started roaming the aisles, scrubbing floors or tracking shelf inventory. The robots that track shelf inventory have proved to be particularly interesting. They're made by Bossa Nova Robotics, a tech startup founded by Carnegie Mellon engineers in Pittsburgh. Their job is to roam the aisles, tracking shelf inventory and conveying that information to people in the store's backroom, on a realtime basis, a significantly faster and more subtle approach than relying on human workers. Some estimates suggest that empty shops can cost a retailer as much as one percent of revenue, a huge abound in an industry that operates on razor thin margins. Should these robots appear humanoid? The company's answer, no. They need to look industrial. You don't want to scare people. Unlike floor scrubbers, shelf tracking robots need to be tall enough to scan all of the shelves from top to bottom. As a result, would they improve intimidating to shoppers? Bossa Nova's solution, make the main unit short and squat but attach a tall arm with cameras to it. How would people react to them? Bossa Nova as well as Walmart discovered that after initial curiosity, customers basically ignored them. What about store associates? Since it's people who must fill the empty shelves, associates started seeing them largely as coworkers and have even given them human names. The biggest surprise, teenage customers. They tend to be naughty. Some try to push them around, others may try to sit on them. Bossa Nova has had to make sure that the robots can withstand possible abuse without breaking down or hurting the naughty shoppers. There are some other questions to which the company's still in the process of figuring out their best answers. Should the robots have turn signals, should they beep if they want to reverse? Should these features follow the design of cars or be totally different? As robots and humans start to cohabit the same spaces and operate in ever closer proximity, questions such as these are beginning to pose important questions regarding the design of robots.

Blockchain

A brief primer on blockchain

Selecting transcript lines in this section will navigate to timestamp in the video

- The blockchain technology achieved initial popularity as a mechanism to record Bitcoin transactions in a digital ledger. Since then, it has also found use in managing supply chains, banking, insurance, and other applications. The term blockchain refers to a chain of blocks. Think of them as digital records that are chronologically linked to each other cryptographically. Like the pages in a book, each block comes after a previous one. Each block contains some information. For example, a transaction between parties A and B, a medical record, or a land title. It also includes a unique and cryptic identifier, a time stamp about when it was created, as well as the unique identifier of an earlier block where some change to the earlier block such as change in the ownership of the particular land resulted in the creation of the current block. The entire chain is copied and stored in a decentralized, distributed, and often public network of computers. The information in an earlier block cannot be changed without changing the contents of all blocks that came after it and a majority consensus of the computers where copies of the blockchain are stored. This makes tampering any record almost impossible. Adding a new block also requires majority consensus of the computers in the network. The decentralized, distributed, chronological, and majority-consensus features of a blockchain coupled with reliance on cryptography make it virtually immune to hacking, fraud, or the destruction of any centralized server. This is why blockchains are viewed as a highly trusted method for storing records.

Case study: Tracking diamonds all the way to the source (De Beers)

- Leaving aside Bitcoins, the diamond industry has become an early and enthusiastic adopter of the blockchain technology. Let's see why. The industry has been plagued by two problems. One, blood or conflict diamonds mined in war zones and sold to finance warlords and their armies. Two, synthetic diamonds passed on as natural ones. Consumers are increasingly unwilling to buy such diamonds. As a result, companies which can prove that their diamonds are free from these problems can enjoy a competitive advantage. In 2003, the United Nations General Assembly established the Kimberley Process Certification Scheme, KPCS, to increase transparency by tracking the movement of diamonds and keep conflict diamonds out of the market. The KPCS mechanism required participants to monitor shipments of rough diamonds and certify them as conflict-free. Unfortunately, the KPCS mechanism proved to be a failure. It relied on self-regulation and self-certification. According to many observers, instead of solving the problem it degenerated into a tool for forced accreditation schemes. There was no way to verify the stated provenance of any diamond. To solve the endemic problem, in May 2015 Leanne Kemp, an Australian entrepreneur, pioneered Everledger, a global digital registry for diamonds powered by an IBM blockchain platform. Kemp's insight? Blockchain can create an immutable ledger to record the entire history of a particular diamond and eliminate the possibility of document forging. Over time, many of the major players in the diamond industry, including miners such as the Russian company Alrosa and Hong Kong-based retailer Chow Tai Fook, have joined the Everledger project. Chow Tai Fook also created the T MARK line, in which the diamonds are inscribed with traceable codes so that their origin could be checked. Everledger is not the only blockchain platform in the diamond industry. In January 2018, De Beers Group, one of the industry's biggest players, launched a tracker with the stated goal of improving the transparency of the diamond value chain by creating permanent digital records for every diamond registered on the platform. Other blockchain-based platforms to track diamonds continue to emerge. Over the past year, Russia's Ministry of Education and Science launched its own blockchain solution for responsible diamond trade. A similar platform called Clara has emerged from Canada. It's clear that the blockchain technology is now becoming an integral part of the diamond industry.

Case study: Digitizing the global supply chain (Maersk-IBM)

- The global shipping industry has historically been plagued with a morass of complex documentation among a large number of parties, including customs and border control authorities who must know the origins, destinations and contents of incoming and outgoing containers. These challenges, the need for trust in the records, the large number of organizations involved, and the complex documentation, make Blockchain an excellent technology solution. Take for example the shipment of flowers from a grower in Kenya to an importer in the Netherlands. The parties involved would include a freight forwarder who puts flowers from multiple growers into a container, customs and border protection authorities, multiple shipping lines, trucking companies, warehouses and terminal operators. The chain could easily involve 30 different organizations and require 200 separate documents. The shipping industry has relied on Electronic Data Interchange, or EDI, technology for several decades. EDI covers a good chunk, though not all, of global shipping. It suffers from some significant weaknesses. The document-based messages are exchanged linearly among companies preventing end-to-end transparency from the producer to the end customer. Also EDI messages are sent in batch format rather than real time. Existing processes, a combination of paper and EDI-based, have proven to be error prone and wasteful. The morass of documentation is estimated to cost as much as 20% of the cost of physical shipment. Because of needed checks by customs and border protection authorities, goods often spend more time in ports than on the ocean. Given the lack of complete transparency, there is also significant fraud, costing shippers an estimated $600 billion a year worldwide. Determined to find a better solution, in 2014 Maersk, the world's largest container and supply vessel operator, initiated discussions with IBM to explore the potential of Blockchain technology for the maritime shipping industry. Blockchain could serve as a universal ledger that would be visible simultaneously to all parties and completely tamper proof. By serving as a single source of truth, a Blockchain could give the needed assurance to shippers, customers as well as government authorities. In 2016, Maersk and IBM initiated pilot tests of a Blockchain solution with Dupont, Dow Chemical and Tetra Pac. The solution included the use of permission Blockchains whereby participants would need permission to participate in a particular Blockchain. This approach was important to ensure that while companies would have access to the records that pertain to their shipments, they would not have any access to information about competitors. In March 2017, Maersk and IBM announced their plan to create a global trade digitization platform for the entire shipping industry rather than privately just for Maersk. It would be run as a separate entity and would be neutral to all parties. August 2018 saw the birth of TradeLens, the global digital platform created by the collaboration between these two companies. With a hundred major organizations, including companies as well as government agencies, already committed as partners, TradeLens has emerged as a long-sought answer to the maritime shipping industry's challenges.