

Digitize Enterprise Assets for Mobility

Enable High Productivity for Professional Developers

WEBINAR TRANSCRIPT

Vance:

And welcome to the breakout session for the Mobile and IoT Summit. This is the session for Progress. And let me welcome back Sumit Sarkar, chief research officer at Progress. Sumit, welcome.

Sumit Sarkar:

Thank you, Vance, for the introduction. Very excited to be part of this series.

Vance:

You know, we're really glad to have Sumit with us this morning. He's a technology evangelist with a focus on technologies that enable enterprises to rapidly adopt new solutions to support digital experience, including -- as we'll see this morning -- a wide range of mobile applications and iterations. He's been working in the data access infrastructure field for more than a decade, with expertise on web, mobile, developers, data engineers, and even data scientists, and he's got a particularly strong background in adaptive, connective, and cognitive technologies as of late. And he brings all of that great experience to us this morning in his session, "Digitize Enterprise Assets for Mobility." He'll look at how to enable high productivity for professional developers in this session, as the demand for digital experiences such as mobility is putting a lot of pressure on enterprise teams and systems -- not just to deliver apps fast, but to deliver innovative, engaging ones -- mobility projects across web and mobile, of course, voice, chat, and even augmented reality. And the other interesting aspect to the session we'll hear for this morning from Sumit is that more and more companies are starting to investigate serverless cloud native architectures to run these types of apps, both for easy scale and better management, as well as for availability of some very engaging services. So with that, just before I hand it to Sumit, let me remind you that you can download the slides as well as access a host of white papers and other downloads. They're all available for you without any registration right under the viewing area. And we'd like to make this interactive where we can, so to communicate with Sumit, there's a quick little "submit

a question" box area right under the view screen. Type in that, we'll get that comment directly to Sumit. And with that, let me now turn to Sumit. And tell us about digitizing enterprise assets for mobility.

(SLIDE)

Sumit Sarkar:

All right. Thank you, Vance, for the introduction. And I do love questions throughout the session. So today we're going to talk about the -- one way to enable high productivity for professional developers. The way we are going to talk about doing that is through digitizing enterprise assets. And we're going to focus on mobility [based on the track?] today.

(SLIDE)

One of the things we want to start with is defining what we mean by mobility. And so mobility means more than applications.

(SLIDE)

So if you think about how applications are evolving -- I guess some analysts and folks say we're in a post-application era, and so instead of people adapting to apps, apps need to adapt to people. And an interesting example around that is, I think, if you think about mail, right? If you checked your email on your laptop and then you might check it on your phone, and all of a sudden, those email providers are now even starting to send notifications to your watch. And so what was just an email application is now becoming almost like a multi-channel app experience. And there's not a single email app anymore. So that's definitely a trend happening around mobility, is multiple touchpoints.

(SLIDE)

And then when we look at things like augmented reality, these are very exciting topics. And we're seeing very practical usage for these -- like, let's say you are a field services technician and you're trying to repair some very expensive piece of machinery. Some augmented reality technology can help to guide that technician through some of those areas -- if there maybe is a new machine or model they've never seen before. And so that's really something happening in a mobility context.

(SLIDE)

And that goes for things like conversational UX. I guess from a user experience, you might be interacting with a chat bot, which are very popular. At Progress, we've launched a chat bot earlier this year. Also, if you're maybe talking to Alexa or Google Home -- enterprises are looking for new ways to meet and engage with their customers, and so these are also becoming very prolific with mobility. Some interesting examples might be mobile workers in a warehouse. They might be walking around; they're busy with tools or boxes or whatever they're working on -- or machinery -- and they might need to ask, you know, how many units do we

have in stock for some kind of part so they can organize their workflow. Or maybe if you want to translate queries, because they have an anomaly detection and prediction solution -- that's an industrial IoT example -- and you've got field workers who aren't actually data scientists and they might just want to ask, you know, what machines need to be replaced over the next 12 months to help plan their day out. And so these kinds of questions can be interacted through the conversational side of things. But the key theme around all these digital touchpoints is that they are often mobile, and maybe there's a remote workforce or remote customer or consumer involved. And so again, mobility means a lot of different things.

(SLIDE)

And when we talk about a lot of enterprise customers, a lot of organizations on this webinar -- and you may not be a startup starting from day one, and so that implies you have some existing systems that are running your core business, and these are often being disrupted by these trends we just talked about.

(SLIDE)

And this is a survey from a MuleSoft connectivity benchmark survey. And, you know, modernization, I guess, has been a topic for the last couple of decades, ever since I've been working at Progress, but I think today it's becoming -- I guess there's additional stress on these systems, and this is why I'm not surprised by this survey. I think this survey was made up of about 650 IT decision makers. I think they're larger organizations, maybe with more than a thousand employees. And so we see that the top concern was modernizing legacy systems, and then if you go on down the line, things like integrating their SaaS apps, investing in mobile apps, migrating apps to the cloud, and things -- so on and so forth. Really, the first answer is almost often the foundation for the next several things in the survey. There are other answers below this, but a lot of this discussion on digital transformation and things like that all starts with making your core business systems and making them ready for mobility and digital experiences.

(SLIDE)

This is another study from Deloitte Insights. And they look at what they call core systems and look at revitalization. This one's from a couple years ago, but similar in intent. I think there's twelve hundred technology leaders that responded. And what was interesting to me is, I guess, the legacy/core modernization -- if you look on the left, the percentage of CIOs that anticipate significant business impact, they didn't believe that this core modernization was a big impact, and yet, that was the top area where the investment's going. And so what this means -- and what I hear from organizations and through my research -- is that they don't really see that as part of the end result; it's really [that?] that's supporting that core/legacy -- core modernization is what supports the things above it, like the analytics, intelligence, the digital touchpoints we just talked about, cloud computing. Things like that -- that's core to it. That's why you see so much investment in those areas.

(SLIDE)

The question then becomes about how do we digitize these enterprise systems to better support those touchpoints.

(SLIDE)

And what we're seeing today is, a lot of existing teams and systems just are not ready for this transformation. And so the delivery expectations are always putting extra pressure on IT and application teams to deliver more rapidly -- you know, and if they don't deliver more rapidly, the consumers get frustrated, the business sponsors get frustrated, and they might start to work with other agencies and start to duplicate efforts -- things like that. And then we see that a lot of custom development has to happen on some of these systems that aren't necessarily ready for new development -- just the sheer amount of load these things add, right? When you start to talk about -- just mobile alone, I think, adds about four times the number of requests on an existing system over -- or just a website. And so when you start to multiply all these experiences, it becomes a scaling problem. And then adjustments to organizations -- if you start to look at mobility, you start to think about continuous delivery and other adjustments to DevOps and how software is delivered -- these things are all fundamentally changing. And so this is what we mean by existing teams and systems are not ready. In some organizations, they've made a lot of progress, so you can't say that across the board, but this is what we see across the majority of organizations that we've done research with or who are our customers -- who started out as a customer, now they are now on that journey. A lot of times we see some of these new requirements -- they start to outpace a lot of these kludgy workarounds, (inaudible) modernization seems incremental and just not keeping up enough.

(SLIDE)

So what this ends up with is what we call "the enterprise experience gap." I guess "experience" means two different things there, maybe. So first of all, there's these visual experiences we highlighted earlier -- things like mobile, chat bots, wearables, smart devices, websites, augmented reality, virtual reality, things like that. And then the developers building on those systems -- they may not know how to work with these enterprise systems that are increasingly important to integrate with from having that single source of [truth?] to your enterprise data -- stuff that -- the systems that run your business. And then also from a security perspective, a lot of organizations have enterprise authorization systems and authentication systems. There's a mix of these things. And so there's this gap where a lot of times these two worlds don't necessarily communicate with each other. People building the things on the left may not speak the language of the interface on the right, or vice versa -- it may be the developers on the right don't know how to build some of these new technologies. And so this is what we mean by "the experience gap."

(SLIDE)

Enterprise system architecture -- this is taking one example. Let's say -- there's a lot of articles out there and analyst reports around ERP modernization. So if you think about a business application -- a lot of what we call monolithic ERP megasuites out there. And these monolithic applications -- they're traded as a single, logical executable. Usually three parts: they have a presentation layer; some kind of server side -- an app server; and then some backend services like a database. When you develop on these, and you do

some kind of an update to the system -- you check in a code change or add a new module -- it requires the organizations and IT teams to deploy a new version or release of this application. Remember, we talked about it being a single, logical unit. So you have to take down the system and make your update, and this can disrupt things. And so this presents this monolithic architecture that's tightly coupled -- so you have your frontend and backend tightly coupled. And it often scales vertically, so when you're building out these clusters to scale, you may have to add servers or add capacity -- things like that. It often scales vertically by adding more -- bigger systems and maybe some clustering, as well. And so in contrast, when you're supporting new, multi-channel experiences, you really need to start thinking about continuous delivery and deployment. Sometimes, organizations are updating on a daily basis, even. And so this is where microservices start to get introduced. And they're needed to deliver agility. And it does this through encapsulated business-oriented APIs. And these can be deployed independently across these digital touchpoints. And it's really based on the principle of loose coupling, as opposed to these tightly coupled systems. And again, this -- mobile's continuous innovation, it really requires more of a microservices architecture, versus what you see in this diagram here.

(SLIDE)

And this is just an example of an architecture -- of what an ERP system might look like. It may have various different pieces -- integration modules and things like that. When you start to look at some of these architectures, it's not always possible to just refactor this enterprise application into a set of microservices overnight. I think that what has to be done -- it has to be done in a gradual way, and I think that's what a lot of modernization conversations are about. There's been modernization done even with this architecture, so it's incremental change, and then when you try to update and improve a system without really trying to replace it. And so I don't think the goal of a lot of these initiatives to modernize is to replace these systems that are running your business -- it's to really, let's say, transform them to better support your application delivery requirements. And not just for, let's say, enterprise business applications -- I mean, it also applies to enterprise APIs, too. They may not support a full range of requirements when you're, like -- for example, if you're building mobile apps. Beyond these systems of records, even enterprise APIs, other business applications -- they're not all designed to support multiple touchpoints, and so we have to start thinking about, How can we modernize more than just systems, but just enterprise data in general as a whole?

(SLIDE)

And if you start to think about -- some organizations say, Just let's adopt some kind of cloud technology to improve our scales. An easy way to do this is to just move an application to the cloud. Some people say, "Just lift and shift." And this is an easy way to decrease some infrastructure costs and get some of the advantages of cloud -- maybe get some vertical scale by pre-allocating some larger machines in the cloud. But it doesn't necessarily provide the benefits of a cloud-native architecture -- you know, things you get like OS abstraction, elasticity, resilience, and things like that. It's a good first step, but -- so I think I went to a DevOps conference, they called it a "cloud-washed" approach. But it's still -- at the end of the day, you get some benefits, but your architecture still remains monolithic, tightly coupled, and it's still going to scale in a vertical way -- and you're going to add more capacity to those virtual machines you're managing.

(SLIDE)

And so when we start to look at this from those new touchpoints we talked about and you start to look at building web applications, you look at mobile and IoT applications -- and maybe, like, cloud applications -- more like other SaaS applications your organization may be running. And so when they need to start to interact with that system, again, those systems don't necessarily scale horizontally. And so that's what we're really talking about with this new engagement's increasing load on these systems. And I think that most IT organizations, they actually limit the access to these systems by design, because those are running their core business, and they can't afford to have unexpected workloads -- like, you know, to set up an Alexa skill to start asking it questions, they don't want to just connect it directly to the existing APIs, because that can affect core operations of the business. I think this is really what the presentation is about -- is illustrating that the majority of these digital projects, they need to run on -- if you didn't have this enterprise system of record -- if everything was starting from day one, then you might start with a serverless, cloud-native architecture. But really the question is, how can organizations still meet these demands for digital experiences on these enterprise systems? And so that's what we're going to talk about. We're going to talk about a couple of approaches here on how you can bridge these systems that are traditionally deployed on servers with cloud-native, serverless architectures.

(SLIDE)

So the first approach we'll talk about in terms of digitizing assets -- and this is a very popular one -- is API management. API management software tools will typically do things like control the connection between an API and applications that use it. They ensure consistency between a multiple-API implementation [and versions?]. They might monitor traffic from individual applications. It can provide memory management and caching mechanisms to improve some of these applications' performance. And they can protect the APIs that you expose from misuse by wrapping in security procedures and policies. And in the past, I think, APIs were usually built and consumed in the same development group, and maybe that group had a good understanding of that domain that API's functionality was required to do. And now with this API economy, a lot of organizations are using these technologies to open up their APIs to new business channels, new partner integrations, and even new markets. So that means their API strategies are really driven to expose data in a more generic way. But it's not necessarily -- as this illustration shows -- it doesn't provide abstraction for application developers. And our theme here is on high productivity. From an application development perspective, the yellow means you can certainly do it; red means you can do it, but there's some shortcomings; and I didn't color it, I don't expect any comments there. But from a web development perspective, you can certainly build against the API management systems -- you can build applications, but it doesn't provide that abstraction layer that -- client libraries and SDKs -- that developers like from a common backend. And then from a mobile and IoT perspective, you can also build mobile applications, but you may not get the features from a lot of these API layers for things like live services, real-time databases, you may not get offline access, push notifications -- there's a lot of series of mobile services that API management may not provide. There's other differences, too -- like if you have contextual data, you need to support logins from those users and to create new users -- there's -- all of those supporting things that are required for a lot of these applications that you're building. And so this is one strategy to digitize data, but we would say that it's not necessarily for high productivity in application development, and there's some cautions there.

(NEW SLIDE)

And then in terms of digitizing those systems, another option -- let's say you're talking about maybe a business application -- you have options to upgrade or a re-platform. And upgrades and re-platforms are usually provided by the same vendor as your business system. And so the example may be, if you're an SAP shop, your upgrades are probably something like -- if you're running SAP on S3, their ERP system version, and you need to upgrade -- let's say you want to upgrade to the latest -- so I think there's -- SAP HANA S/4, I think is the latest version, and it runs on that -- on the SAP HANA in-memory platform. And that one is more cloud-ready, or you could also run it on premises. But you could certainly do upgrades and digitize some of these things, because you get a more modern architecture. And then you can also re-platform (inaudible) migration, move assets. For example, you want to move some things to the cloud with a smaller amount of upversioning -- like, maybe you want to move a managed database offering or part of your system, or maybe add some Autoscale-enabled scaling and things like that -- that way, you can benefit from cloud infrastructure. We've seen that, as well, and have also seen organizations look to leverage -- like, maybe some NoSQL databases to try and provide some scale on top of their systems. So re-platforming -- there's a lot of different things you can do and incrementally work on it -- work on improving the scale. From the perspective of the diagram, a lot of these things require you to change and disrupt your business processes by impacting those systems by doing an upgrade or re-platform -- not always the best option. And then a lot of those systems, even with re-platforming, they still don't provide a lot of the mobile services developers are looking for. And then from the cloud -- third-party integrations from the different SaaS vendors, sometimes it depends on what you're doing here. Sometimes they can work seamlessly, and sometimes they don't have integration -- or the re-platform may not be set up to be accessed from external systems. There are some cautions here, as well, in terms of high productivity for building some of the digital applications.

(SLIDE)

And then when we look at integration platforms as another option -- you look at the iPaaS acronym -- it really connects in the software applications. It might be deployed in different environments -- things like cloud -- maybe cloud and on premise or cloud to cloud. They're really popular to synchronize data, but I wouldn't say they're necessarily also engineered for presenting an abstraction for application developers to be rapidly productive. You can certainly build those apps, and I think I've seen some of the iPaaS vendors start to provision APIs and become more like API management vendors, so there seems to be some blended functionality in a lot of these use cases. But the end result, again, is -- it will provide integration with cloud applications, so iPaaS can synchronize data between on premises and in the cloud. And so these are three approaches that we've seen organizations try. And certainly, they can get a lot of apps delivered, and they provide incremental progress, but I think that from the perspective of an application developer --

(SLIDE)

-- and an application developer who's looking to build digital experiences and in a highly productive manner, we believe that Progress's serverless cloud is the answer. Enterprise engagement is the delivery of what serverless clouds can provide.

(SLIDE)

And so when we look at Progress Kinvey and enterprise engagement -- we talked about these new digital experiences -- wearables, mobile applications, watches, new form factors, chat bots, smart devices -- all those -- it's exciting stuff, and Progress has different initiatives around a lot of these different touchpoints. But when we look at it from the perspective of enterprise data -- and serverless clouds sit on top of enterprise data and provide an instant engagement layer that really reduces the risk and cost in terms of development resources and business interruptions from completely rewriting some of those on-premises systems or re-platforming, re-architecting, and things like that. And with the serverless cloud in place, we really get to a point where you can move from this maybe monolithic architecture that doesn't necessarily scale and you move to a cloud-native architecture so you can start to add as many touchpoints as you want. You can start to do your voice and start to talk to your SAP systems; you can start to get notifications from your different devices. And so that's really exciting to a lot of enterprises -- to increase enterprise engagement. You also move -- to a tightly coupled architecture, something that's more decoupled, and so you can start to take advantage of having some of these microservices environments and have more continuous delivery of new features from your serverless cloud layer. And again, we talked about scale, as well, right, to support these touchpoints. So I think that when we talk about putting this [instant on?] and serverless cloud on top of enterprise systems to become that engagement layer for those touchpoints, it's future-proofing a lot of future things you may not be thinking about today. And so this layer enables future -- let's say -- it's kind of like a modernization layer that is future-proofing -- it doesn't necessarily touch your enterprise data. And I think that's the biggest point. The most important aspect of this for a lot of organizations is not to interrupt that enterprise data, and so this is the architecture approach. And let's dig into that a little bit more.

(SLIDE)

So we call it "no touch" serverless cloud backend, meaning that you can deploy the Progress Kinvey serverless cloud backend and it runs in the cloud. And it runs in our cloud -- we host a serverless cloud backend -- or you can deploy it to your own clouds. If you're Amazon, Azure, or Google, you can deploy this to your own clouds, as well. With this, you get native mobile functionality. So if you look at the web development and mobile development, there's no more cautions around development, because you get things like offline support, data encryption, user management, live service for mobile apps. You get a common app backend across both web and mobile. And again, this enterprise data -- the things down on the bottom -- I put a couple of things down below -- behind the perimeter network for -- maybe it's your organization or maybe you have a cloud-hosted version of one of these applications, but whether you have an ERP system or business application like SAP -- from SAP, QAD or Oracle -- or maybe you have, like, the enterprise API layer -- maybe it's IBM, maybe MuleSoft, or maybe some databases that you want to expose to some of these new experiences. And so these are the enterprise systems that you can expose and abstract for developers, who get -- the serverless cloud backend includes SDKs -- so client SDKs. So they communicate all this stuff through a common protocol, so that's how they're increasing their productivity. And when you look at some of these things like microservices, you can start to create business logics for your applications. You can write in JavaScript on our Node.js platform. And you can write and deploy them as microservices. And what this means is, you can refactor your applications over time -- and to start to

support microservices and continuous deployment of new features, without touching your enterprise data systems -- whether APIs or applications or databases. A lot of people look at this and you see a lot of things happening -- you see this common presentation layer. And so again, this is all about productivity for application developers. So in contrast, again, to API management -- you know, for example, enterprise data is really abstracted and for specific consumption through those SDKs. And so if you're working with the -- one of the most common interfaces that we see people connecting to the server -- serverless cloud backend is to actually an enterprise API system. A lot of those organizations running something like MuleSoft will leverage this to the front to provide that developer experience and mobile services for their applications. And a lot of them also look to augment data -- so they might be building applications on some of the MuleSoft APIs through these SDKs, and then they might need to do some lookup and maybe merge some data from, like, Salesforce or another cloud application like Workday. You then get to -- you can build this business logic, and you end up building a rich application data layer for your developers to use across your both web and mobile applications. And so this is how the serverless cloud is delivering productivity. So it's digitizing one or more systems in a developer-friendly way.

(SLIDE)

And to break down some of those services -- there's a lot of things on here, but if you look at -- think about from an application development perspective, where you get -- the services include things like -- if you think about the client experience on the device side or on your mobile devices and mobile phones, you get things like the data caching for offline support. And then once you get back online, there's synchronization and resolution. And then from a performance perspective, some of these enterprise systems may not want direct access, so we provide cloud caching so that a lot of that load is handled by the serverless cloud tier. There's things like, like I said, the live service; again, things like a real-time database if you want to -- your phone always gets the latest updates from the backend systems. You have access controls, operational intelligence, and things like that. So there's a host of different services engineered to provide the digital touchpoints with the services they need to build those applications that aren't otherwise available in enterprise systems. And so we expect this to rapidly increase productivity for developer teams. And then I think on the backend, if you think about -- we talked earlier about mashups of enterprise data sources. You might want to start building queries that are application-specific to deliver the right data to the application; that you may not want to disrupt enterprise data models and you can build your own application data model. There's also things that are contextual to your applications -- maybe files, user preferences, things like that. And again, the engagement features, like push notifications, analytics -- all these are more on the backend side of these cloud services.

(SLIDE)

And to give a customer example, the director of SAP applications at Bell & Howell shared this quote in a press release: and “if you’ve worked with SAP systems as long as I have,” with that cloud caching, “the performance has been unbelievable. I went from 14-second SAP data access down to 400-millisecond mobile-friendly data access.” This is not necessarily to say the SAP is a slow system. The expectations of your mobile users are ones where they’re not going to wait for the latency, potentially, of integration or an API call. They’ve got a mobile app; consumers are impatient; some of my colleagues are very impatient, as they want their data quickly and they want a native mobile experience. And you can deliver that on top of SAP with Progress Kinvey

(SLIDE)

(inaudible) a lot of different approaches to digitize enterprise systems. For some organizations, there’s a lot of these that work great, and then for others, they may not.

And so from the perspective of productivity, I’ll give another customer example. And so with Canopy Health -- this is another -- a public reference -- they adopted Progress Health Cloud, which is built on the Progress Kinvey platform. They have backend systems -- like, IBM Initiate, I think, is their master data management service, and they use Salesforce for identity management. So they have some of these enterprise systems we’ve talked about before. And they are a patient-centric healthcare provider, and they want to focus on improved experiences. A lot of things with these complexities with the systems and processes made that a little bit more challenging. And so to streamline some of these experiences across mobile and desktop platforms, it was really important to deliver that patient-centric experience that they wanted. They leveraged the Progress Health Cloud, which is part of the Kinvey serverless cloud backend solution, and they observed a cost savings of, like, 70 percent. And in addition, this serverless platform -- they became agile and really sped up the development. I think they [excited?] from, like, 18-week cycles to, like, 6 weeks with the technology. So you can see in this use case the things we talked about -- they’ve got some enterprise systems, things may have not been as fast to build web and mobile applications, but then they leveraged the serverless cloud platform and came up with this common application abstraction layer, and they’re able to deliver web and mobile applications to deliver these patient-centric experiences. So this is a good illustration of what we’ve talked about.

(SLIDE)

And then when you look at productivity -- when you start to look at agile teams and you look at this serverless cloud backend -- to deliver web and mobile applications, you might have needed a larger team to deliver one application. And so let’s say the mobile services, the application abstraction layer, digitization -- you can deliver the same level of quality applications with a more agile time, right? So you can have one team of developers. And our SDKs allow, if you’re a developer in -- if you’re a JavaScript or (inaudible) Angular developer or a .NET developer or whatever developer you are, we enable you to build applications and deploy across multiple touchpoints, and you can reuse code -- and general productivity gains you get. And so you can be more agile with smaller teams. And then the fact that this is a serverless cloud backend

-- the cloud service and DevOps resources are more productive, because they have a lot of the infrastructure and scaling happens on our service. And so these teams are more productive, and you can start to quickly deliver these applications and really accelerate the pace of delivery across your organization, whether you're an application development partner, digital agency, or as an IT team delivering an application delivery.

(SLIDE)

As I talked about, the multiple touchpoints and reuse code and skills -- when we look at a journey with this serverless cloud application platform, you may go from -- you build a responsive web application; you might go to a progressive web app; and then all the way to native. And we support the full journey for applications, as you deliver based on your user requirements. And we take care of all the stuff on the bottom, including digitizing assets. And your teams can focus on user experience, the business logic that matters to your organization, and we take care of the rest.

(SLIDE)

And again, once you digitize some of these systems, you get access to -- [we'll say?] abundant developer talent -- I did a search on indeed.com; if you were to search for a developer in the skills of your backend enterprise system, you'd get a smaller number, but if you search for a JavaScript developer, with a serverless cloud backend, you have access to more developers -- like, 28,000 jobs. In my research, I've met a lot of organizations that say, We have really ugly enterprise systems and we have a lot of security requirements and we can't attract the talent we need. And our serverless cloud platform's changing that discussion, and they're able to present things in a way that is compelling to attract new talent and then really focus on innovation and not on this new integration with enterprise systems and things like that.

(SLIDE)

I really appreciate you sitting through this one, and I'd love to hear from you. You can contact me directly at any of these -- email, you've got LinkedIn and Twitter. If you want to learn more about our platform, you can visit progress.com/kinvey.

(SLIDE)

And I am very interested in connecting around this, because I think the message of developer productivity is very important to myself in enabling developers to be productive on these enterprise systems. And we want to hear from you if this is of value -- feedback. And also, I want to hear from you if you think that -- you know, if you've [solved?] this some other way, we would love to hear from you. That wraps up our session, and I will turn it back over to Vance for the Q&A.

(SLIDE)

Vance:

Sumit, really great session -- really multi dimensions going on here, but one of the things I liked most about it is that aside from developers, we've got the architects that worry from a big picture how to get rid of the mobile and other app backlogs, as well as the IT ops and the guys that have to run the apps once they're built. So, a really great session. I'm glad you were able to touch on so many different personas.

(SLIDE)

You know, Sumit, you mentioned the API toolkit. I think a lot of people in mobile world especially -- that is the go-to toolkit for doing all sorts of stuff -- not just data import and integration, but even some sort of early stages of moving from design to pre-provisioning and tests. I wonder if you could give us the big picture -- as you describe this to your customers -- this migration to the serverless cloud -- where are the touchpoints? Where are the places that they can actually get some gravity to begin that shift from a simple API environment, which may not scale and may not give them the other advantages, into this new serverless cloud -- especially, as you said, with this idea that you don't want to break or slow anything down while you're making the move?

Sumit Sarkar:

In terms of the touchpoints, I think that we talked about -- there's a lot of different strategies to digitize enterprise assets. And the one that our partners and customers continually turn to is one that you've mentioned -- you hit the nail on the head -- that limits the disruption and risk to existing systems. And so that's where you're able to leverage and deploy this serverless cloud platform that sits on top of your enterprise data. And so that provides the scale and the application abstraction that you need to build out those touchpoints. And again, we contrast it with several other options, and there's this journey to a lot of this modernization. So once you get started with this serverless cloud approach, then start to add on microservices, business logic, and really evolve your application abstraction layer, and then you can start to serve multiple touchpoints and future-proof that strategy.

Vance:

You know, Sumit, it's such a great high-level picture.

(SLIDE)

Because there's a question comes in that talks about -- kind of at an implementation level -- and it simply says: From the Progress presentation, I get the sense there is technology, but also a methodology that goes along with helping us move to the serverless cloud. Could the speaker detail a bit about the real key platform features, as well as the best practice or the methodology steps that would help us adopt this quickly?

Sumit Sarkar:

A good question. When you think about some of these digitization approaches, a lot of them are not something a mobile and web developer can take on themselves. And so with the cloud and the advantages of cloud, you can adopt and innovate with a serverless cloud platform. You can go up to our website and you can actually just register for a trial and you can get started from day one. As a developer, you may not know that, you know, I want to go and -- I don't know how to build APIs and publish them through, like, a MuleSoft, or I don't know how to wire up all of these systems. And so the promise of the cloud is the productivity, right? You get to leverage cloud technology, you get to provide this engagement layer, and then you get to start building apps in the language of your choice -- like, using the SDK of your choice, and as abstract as -- you can work with OAuth authentication versus, like, learning LDAP and other things on the backend. Or you can start to maybe work with your SDK -- you write a line of code rather than having to learn different API concepts to negotiate, authenticate handshakes. The cloud provides instant productivity for developers. They just sign up and get started.

Vance:

You know, Sumit, you mentioned developers, and here's kind of a hands-on question.

(SLIDE)

Certainly, the word -- or the company -- Kinvey is very familiar to folks in the mobile space for any amount of time that have been working with MBaaS or regular backend as a service platforms. And just to note here that Progress acquired Kinvey last year about this time -- around the summer. So it's kind of an interesting update on the general space. And the question says: Is the Progress Kinvey platform similar or different for delivering high productivity for some of these backend as a service platforms?

Sumit Sarkar:

Yeah, that's an interesting question. When we think about where Kinvey started, there's hundreds of thousands of users across web and mobile accessing data through Progress Kinvey. When we take Progress's best of breed frontend tools, Progress is also the creator of NativeScript, the open source JavaScript cross-platform framework, so you can build applications and they deploy natively to Android and iOS. And so there's productivity across backend and frontends that would combine Progress's frontend and backend technologies. And there's a lot of productivity -- a lot of them focus on, let's say, class of developers -- maybe they can call themselves [citizen?] developers -- and they're productive with that platform. And what Progress is looking at is a high productivity; what the class of developers are looking at are more about web and mobile developers, so they can leverage their existing skills and deploy to multiple touchpoints and be productive. We're in the same space; we're in the same end goal; we just serve different classes of developers.

Vance:

You know, Sumit, so many different capabilities here with both the platform and the serverless cloud. Kind of

a one-plus-one equals three question comes in.

(SLIDE)

And it simply asks: I'm very impressed with the combination of services and functionality involved. Can I create function as a service against my enterprise systems using Kinvey and a serverless cloud?

Sumit Sarkar:

Yeah, that's a great question, and that's really the promise of serverless technologies, right? It's one of the aspects that's popular -- is its function as a service. Progress Kinvey platform has flex services, so you can build out these functions -- and you can create event-driven architectures, you can [call?] business logic enterprise systems. And so effectively, you're getting this very scalable -- to get in a cloud scale of these functions of service and serverless computing. But again, it's all done against any enterprise systems, and that's very compelling to a lot of application developers. And if you look at, like, the Stack Overflow 2018 survey, I think serverless was the second -- or was it the most-loved platform out there by developers. With this architecture, you can call it in code you're familiar with, you can invoke business logic, you can start to build some of these really modern [architectures?]. But the freedom of choice for developers is really part of the benefit of the serverless cloud.

Vance:

You know, Sumit, time is just about up I see, but before we let you go, give us a sense of what the next step would be.

(SLIDE)

We've talked about both the high-level technology as well as some implementation -- I think you mentioned a trial a couple of times here. Give us some very tangible next steps you would recommend attendees take to learn a bit more about what Kinvey's doing to help get into that serverless environment.

Sumit Sarkar:

Yeah, I would encourage you to take a look at some of the assets. We have some papers, some analyst reports; assets on our website, as well. And the biggest and most exciting thing, I think, is the trial you can sign up for and get started building applications. And the trial is a developer version; it's free forever, and you can play around as much as you want. And then if you need support or you're ready to build an app and deploy it, then come talk to us. We have a pricing page on our website to understand what the pricing and levers look like for the different versions.

Vance:

Fantastic. Fantastic. Sumit Sarkar, chief research officer at Progress Kinvey, thanks very much for a great

session -- and a really clear road map on how we can get into the wonderful, productive world of serverless cloud computing without having to go back and get our PhDs. It was a really terrific session -- especially for all of the mobile folks out there that are working hard to support all these new modalities you talked about.

Sumit Sarkar:

Thanks, Vance.

Vance:

As Sumit mentioned, we've got several of these assets. They're here for you right now. And let me just remind you, they're available without registration required. The PDFs will download directly to you without having to fill out any form. If you go to the free trial, you will have to sign up there with Progress Kinvey, but it's a very rich environment. In fact, as Sumit said, it's free forever, so we don't see that very often.

(SLIDE)

And one last note: Progress and their team had given us some other links. We didn't have room for them all, so we put them here on this slide, all in one place. These links will be live, so the resources will be available to you once you download the PDF copy of Sumit's slides. Thanks again, everyone.

END OF AUDIO FILE



About Progress

Progress (NASDAQ: PRGS) is a global leader in application development, empowering enterprises to build mission-critical business applications to succeed in an evolving business environment. With offerings spanning web, mobile and data for on-premise and cloud environments, Progress powers businesses worldwide, promoting success one application at a time.


Learn about Progress at www.progress.com or 1-781-280-4000.


Worldwide Headquarters


Progress, 14 Oak Park, Bedford, MA 01730 USA

Tel: +1 781 280-4000 Fax: +1 781 280-4095

On the Web at: www.progress.com

Find us on  facebook.com/progresssw

 twitter.com/progresssw

 youtube.com/progresssw

For regional international office locations and contact information, please go to

www.progress.com/worldwide

Progress is a registered trademark of Progress Software Corporation and/or one of its subsidiaries or affiliates in the U.S. and/or other countries. Any other trademarks contained herein are the property of their respective owners.

© 2018 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

Rev 2018/07 | RITM0016998

