

# The Role of API Management in Digital App Strategies

Carl Lehmann & Sumit Sarkar

---

WEBINAR TRANSCRIPT

## **Colleen Milton:**

Hello, everyone. This is Colleen Milton, and on behalf of Progress and 451 Research, I would like to welcome you and say thanks for attending today's webcast, titled "The Role of API Management in Digital App Strategies."

(SLIDE)

Leading off the discussion today will be Carl Lehmann, who is principal analyst in architecture automation and integration security at 451 Research. Following Carl will be Sumit Sarkar, who is chief research officer at Progress.

(SLIDE)

Just a few housekeeping items before we get started. To ask a question, simply type it in the question box on your screen below. We will answer as many questions as we can during the Q&A session, and we'll respond to all unanswered questions via email after the webcast concludes. This presentation and slides are available for download. And finally, please provide feedback at the end of the webinar. And with that, I'll turn it over to Carl.

(SLIDE)

## **Carl Lehmann:**

Thanks, Colleen. And thanks, all, for joining us today to talk about the role of API management in digital application strategies. For those of you who are unfamiliar with 451 Research, we're an IT research and advisory firm founded in 2000. We help clients -- enterprise clients, IT vendors, and the investment community -- in understanding market trends and technology trends so that enterprises can make better decisions choosing technology, investment folks can make better investments. And we assist vendors with competitive positioning in the market. We have over

a thousand clients globally. We operate in 11 countries. Our hundred analysts track over 2,000 vendors, publishing 4,000 or so reports on an annual basis. And my role in 451 Research is that I'm part of the AppDev, DevOps, and ITOps team here. I focus on enterprise architecture, process automation technologies, and integration technology.

(SLIDE)

And Progress has invited me today to talk to you about the role of API management in involving application development strategies. And the reason for that is because if any of you have been following the market closely over the last number of years, API management and application development architectures have been converging and merging in recent years. So for example, MuleSoft was acquired by Salesforce; Apiary, an API management design vendor, was acquired by Oracle; Apigee was picked up by Google; earlier on, Epiphany was picked up by Microsoft, Akana by Rogue Wave, 3scale by Red Hat, Layer 7 by CA. So you can see that API management has become integral to a broader set of capabilities to enable digital application strategies. So what I'd like to share with you today is the business drivers that are kind of making this happen. And that means I'm going to start out at a 40,000-foot level, to talk about digital transformation strategy and new competitive advantage. I think that it's important for all technologists, and certainly businesspeople, to understand some of the driving forces motivating the changing nature of application development and the role of API management. Then I'm going to kind of come down to 20,000 feet and go into some of the technology drivers, talking about new approaches in some of the enabling technologies that can facilitate next-generation digital application strategy and the use of APIs -- and then speak very specifically about the role of APIs. And then we're going to hand it over to Progress. Summit will share with you their approach to enabling improvement in application development in the current market.

(SLIDE)

So with that, let's get started with the business drivers.

(SLIDE)

No conversation from an analyst today would be complete without a discussion of digital transformation. It's on the tip of our lips just about everywhere we go. People may be a little overrun by the discussion, but fundamentally, the definition of digital transformation is to use new enabling technologies to improve the customer experience, improve business operations, or maybe open new markets and create new business models. And while that's all well and good, there's nothing new there. We've been using new technology to improve the way we do things and improve the customer experience and enable operating efficiencies for years -- for decades. There's nothing new there. What's new in this environment is it's just not information technology we have to deal with. I remember the days where all we had to worry about was upgrading our earpiece system, linking it to our CRM system, and figuring out how to use the internet for marketing. That seems relatively simple by comparison today, because we're dealing with a whole host of other sets of technologies: artificial intelligence, machine learning, IoT -- internet of things -- capabilities. And plus, things that don't necessarily rely on information technology, but operational technology -- to improve the way things are manufactured, to understand our business environment better through things like drones or ground-

penetrating radar or various robotic applications. So it's more than just IT; it's also OT, and all the various technologies that are coming to market to change the way just about anything can be done. That creates a level of complexity we need to understand.

(SLIDE)

But what we need to understand first is what companies might be trying to do with all this technology. Indeed, they're trying to exploit technology to do new things, but what they're really trying to do is go after your customers or go after customers that they didn't have. There's new, aggressive rivalry in each market that we face. And there's new market entrants that can enter a market with no technical debt and become serious competitors to the established markets that we've spent years nurturing. What these companies do -- aggressive rivals and new market entrants -- is they disrupt the market. They change the way business is done by changing the decision criteria in the mind of the customer. They empower customers to enable new ways to evaluate and make decisions; they expose more information, more choices, and provide them tools and recommendation engines to come up with the best conclusion. What this means is, all enterprises in all industries need to be aware of their rivalry -- both from established competitors and new market entrants -- and have an intuitive understanding, as best as possible, as to what customers are expecting. So what that means is, we need to create new competitive advantage all the time. And I want to differentiate between competitive advantage and competitive differentiation, and I'll get into that in some more detail. But new competitive advantage is an imperative for all enterprises. And again, I mentioned earlier, this is a 40,000-foot approach, but I believe that any technologist listening to this discussion should have a business sense of how to understand the environment in which they operate so that they can develop next-generation digital applications serving their customers in a better way to enable new competitive advantage -- and that means a thorough understanding not only of the technical side, but of the business side. And I'm going to go into the business side in a little bit more detail.

(SLIDE)

But that doesn't mean that we just redesign our business processes. It doesn't mean that, oh, we have a new market entrant, we have a new rival, they're getting more aggressive, our customers expect more from us -- that doesn't mean we just redesign the processes and the systems we use to buy, make, move, and sell things. It's not enough to go through a business process reengineering effort or a business process design effort. We have to think a little more strategically -- not only as business professionals but as IT professionals -- to become business technologists on both sides of that equation -- business and IT becoming business technologists. And the way to do that is to use some fundamental principles that have been used in enterprise architecture sessions in programs across the board that date back several decades but nevertheless apply equally well today.

(SLIDE)

One of those mechanisms is what's called the adaptive loop. This is where an enterprise needs to do a much better job and be much more speedy and efficient at sensing change, risk, or opportunity, both from what rivals are doing and what customers expect; interpret it properly, so that you've got the right information,

the right variables, the right data to make the best decision that's possible, based on rules, policies, the counter-measures that you might need to apply to aggressive rivals or the features and capabilities that your customers are expecting from you; and then act -- and act efficiently. What most technologists don't necessarily practice on a consistent basis is understanding how quickly an enterprise senses change, risk, and opportunity, makes the interpretation, decides what to do, and act. There's some flexibility. There's collaboration that needs to be enabled in that process -- in that adaptable process. It's not just structuring a workflow and then executing the workflow. Sometimes, that workflow needs to adapt to changing conditions very rapidly. And so understanding the potential changing conditions and thinking through what might happen using this adaptive loop process, I think, is of high value -- not only for businesspeople, so they understand their overall operations and how to respond, but also IT people, on how they use technology to codify an adaptive digital application.

(SLIDE)

As we try to understand how this is coming about in the market, 451 Research conducts a greater number of surveys. We have a community of 50,000 or so IT professionals on a global basis that we reach out to and ask them questions about various things. In one of our surveys, we asked about their digital transformation strategy. And this slide represents kind of where their priorities were. There are enterprises that don't have a formal digital transformation strategy, and there are those that do. Those that do have formal strategy look at -- 42 percent of them believe a personalized experience is a high priority for their digital transformation strategy; 49 percent believe that it's a contextual experience and contextual recommendations is a high priority for digital transformation; and 37 percent believe that their business processes need to perform and improve and become more adaptive. So this is kind of where some of the priorities are for those enterprises that have a formal digital transformation strategy that seeks to improve customer experience and react and adapt more quickly to the actions of rivals.

(SLIDE)

When you start going a little bit deeper and asking those who, with a digital transformation strategy, how do they go about doing it -- well, they fundamentally have somewhat of a four-part strategy. So the four-part strategy really includes the digital experience, where they have a mobile and cloud-first approach to application development. They have a digital organizational approach, where they adapt some of their culture to think more clearly about the actions of rivals and customers and are doing the best they can to digitize their business processes -- as many as is feasible -- and automate them. They are looking to a platform that's adaptive and flexible enough to be able to enable a mobile and cloud-first application and improve business processes. And they want to expose digital operations in a contextual way, so that they're learning from their experience. And so this enables competitive differentiation. The folks with formal digital transformation strategies that we've been talking to do so with business process improvement and personalization and contextualization in mind. And they think through it with an integral four-part strategy of experience, organization, platform, and operations. And they've been doing a very good job at this -- exposing these next-generation applications to the digital world, as it were. And they've been doing a good job of creating competitive differentiation -- they're different from their rivals. But that doesn't necessarily mean that you win in the market. Being different doesn't mean you're better -- it just may mean you're different, and it could

be in a bad way. What's required is competitive advantage. And when I say "competitive advantage," to many people, it's become somewhat of a vapid term -- it's fallen on their deaf ears -- with no meaning -- because it hasn't been explained in any way. And I think it's of great value to understand how competitive advantage is achieved, and so I want to explain it to you a little bit.

(SLIDE)

When you examine and you study all of the research and experiences and investments made by successful Fortune 500 companies, you look at the research by the major MBA schools and strategic think tanks, you'll discover that there are many ways to create competitive advantage. But all of those ways have a common denominator. It starts with examining how you do things. So the most popular work in competitive strategy and competitive advantage has been performed by Michael Porter, a Harvard professor, who gave us the value chain several years ago and explained it in detail in his books, *Competitive Advantage* and *Competitive Strategy*. But what the value chain does is, it provides an enterprise a means to examining all of the activities that they perform and how they interact -- the activities and linkages. And that examination allows you to understand sources of competitive advantage. And that competitive advantage is achieved, then, by doing the same things as rivals, but differently -- and/or doing different things that that are acknowledged by customers and prospects as superior -- and as a result, they award you their business. So that's kind of the common denominator across every enterprise in every industry of how to enable competitive advantage -- doing the same things as rivals, but differently, or doing different things that are acknowledged as superior, and therefore you're [awarded?] your business. And the other way to think about that is how you do things. And when you think about how you do things and then you think about the last slide I presented, where companies are very interested in improving business processes and contextual personal experiences, that comes down to how.

(SLIDE)

So when you look at what I said earlier about rapid sense and response and you combine that with the value chain -- when you have a thorough understanding of your activities and linkages and how you can uniquely position relative to rivals that's acknowledged as superior -- doing so by enabling a rapid sense and response set of capabilities -- how quickly do you sense change, risk, and opportunity; how rapidly and accurately you interpret what needs to be done; how quickly and efficiently you do it -- decide to do it and then do it -- when you combine that kind of thinking -- that analysis -- with a value chain analysis and looking at how you do things differently, you're able to create competitive advantage -- something more than competitive differentiation. So that's the strategic thinking that I wanted to kind of set in place at a high level. But the next question is: okay, sure, Carl, how do you do that?

(SLIDE)

Well, when we go back and examine our survey community, they had an integral four-part strategy: digital experience, digital organization, digital platform, digital operations. Well, an integral strategy needs an integral environment.

(SLIDE)

So with that, I'd like to share with you a little bit more about the technical drivers. So now we're coming down from 40,000 feet; we're coming down to 20,000, maybe 15,000 feet to get a little closer to how application development is affected and how that ultimately plays into the value and role of APIs.

(SLIDE)

This whole environment that we operate in is driving and requires new approaches to what I refer to as a process-oriented application development environment. Historically, in the past, many of the applications that IT organizations built were mostly for employees. They were built in integrated development environments or service-oriented architectures that were good, but they were complex and costly. Waterfall methods are still kind of practiced a little too often today. But they were systematic but slow, and they were built on monolithic, data center architectures. That's changing. The diagram here, you'll see, kind of demonstrates a two-year shift of how enterprises believe their application environment is going to change away from the data center and onto software-as-a-service and infrastructure-as-a-service. And so many enterprises now are practicing modern, agile application development techniques. The business teams and the IT teams are moving together to rapidly prototype and iterate new ways of doing things -- new business processes, new user journeys, user experiences. And more and more, applications are developed for customers and partners -- external parties, not just the internal employees. And they run across multiple execution venues, which require companies to think about, What's the best execution environment that I need to run a specific workload, a specific application? How do I maintain that? How do I get them to interoperate with one another? Et cetera.

(SLIDE)

So what's needed now is a new thinking around AppDev and DevOps, where development and operations is getting pushed together. And the next generation of technology is coming to form what I refer to as a digital automation platform and hybrid integration platforms. So in our surveys, across the board, in many instances, many enterprises prefer cloud native application development to get started with new applications and are practicing lift-and-shift to get to cloud architecture very quickly, as I'm sure you're all aware. They want to, on an accelerating pace, exploit shared cloud architectures, but also containers and microservices, and how they can containerize applications, exploit the rapid connectivity theoretically of multiple microservices to execute logic -- and oh by the way, every container and every microservice communicates with what? An API. We'll get into that. The development of applications and their deployment to productions can happen automated -- in an automated fashion -- almost instantly, through CI/CD -- continuous integration, continuous deployment. The operations and development silos are pushing together -- I wouldn't say they've converged, but they're pushing together, where development and operation teams are working side-by-side more often than not. And the tools that they're using -- the [tool of chains?] that they're using -- are either converging or linking in some way to share data. Many enterprises have procured business process management suites. When a SaaS capability was not available, BPM suites allow you to build your own business process applications. At the same time, they've been kind of evolving to include low-code and no-code capabilities. Platform-as-a-service from hyperscale vendors have very sophisticated, highly scalable application

development environments. Both BPM suites and PaaS environments are adding policies and rules as an integral set of capabilities, so that process, performance, and execution can be better tracked, and business outcomes can be managed better. The use of low-code/no-code is a natural evolution of next-generation software development to make things more intuitive, to make it easier. And everyone is experimenting in various ways, shape, or form, with machine learning and artificial intelligence. And when you combine all that stuff together, basically, what you're capable of building is intelligent process automation -- so you can get smart of how things work, you can [get?] -- expose feedback loops to determine performance and execution, outcomes and results, feed that back into a design environment, and enable what many companies are calling -- including myself -- intelligent process automation. And what that means is a next generation of technology stacks is starting to converge into automation platforms of various types.

(SLIDE)

So when you think about digital transformation, you have to first think about the user. What's the journey that you're on? First of all, who is it? Is it a customer? Is it an employee? Is it a partner? The journeys might be different. So: who are they? What are they doing? How do they need to do it? What's the device that they use? What kind of user interface is necessary? What applications do they require? What data do they require? What processes do they execute -- or processes that they trigger? What's the infrastructure that best supports these environments? What services do you need to support them? And does this affect partners? So that kind of thinking is the logic that's got to go into your value chain analysis and your adaptive loop analysis. How quickly can you sense change? How quickly can you interpret what needs to be done? How quickly can you make the decision to act and then act -- through understanding how you adapt your value chain in response to risk, change, or opportunity?

(SLIDE)

But that means you need some stuff. So this is thinking through the processes and the logic. How do you design and model new processes and logic? How do you enable configuration and orchestration? What kinds of rules and policies and KPIs need to be codified, and how do you control events and analyze execution and performance and outcomes?

(SLIDE)

Application development -- this is where you get into the coding. What kind of tools do you use? High-code, low-code, no-code? A lot of enterprises are doing a combination of all of the above. A package is built and deployed and it helps with provisioning and configuration management. These things need to be thought through in terms of how efficient you are.

(SLIDE)

The next step is, how do I integrate all of this? Integration-platform-as-a-service is very popular among enterprises for cloud-to-cloud and cloud-to-ground integration. It uses connectors. But it also has more sophisticated capabilities -- message queuing, enterprise service bus, and sometimes managed services bus.

But in the heart of that is API lifecycle management, and that's going to be where I'm going to start talking in a little bit more detail in a few minutes.

(SLIDE)

Next is the data. Where is it? What kind of quality is it in? How do I aggregate it? How do I expose it? How do I enrich it? How do I translate? This is the kind of capability that needs to be integral to a lot of the development platforms we use.

(SLIDE)

And then backend services. There's enrichment and value-added services that we can exploit to better improve customer experience and react to rivals -- and things like various SaaS applications, social media, and some more complicated IT architectures that allow us to lower our cost of provisioning and deployment.

(SLIDE)

And then finally, no marketecture like this would be complete without a discussion of security. I'm not going to talk in detail about security -- I'm not a security analyst -- but any platform needs to have a discussion of how they control authentication, authorization, and identity management through a policy-based system.

(SLIDE)

Okay, so now we're going to land on the APIs. So what does all this mean, and why is API management so important?

(SLIDE)

Well, in a lot of cases, API management has been overlooked by enterprises -- in fact, in many cases. APIs are not new -- we've been developing and using them for years. The problem has been that they've never been looked as an asset -- they've never been looked at strategically as a product. That's changed in recent years. APIs are very valuable assets to an enterprise. So when you look at a technology stack like that, you say, How many APIs actually enable something like this?

(SLIDE)

Well, when you kind of squint a little bit and look at the API complexity -- wow. There's a lot of stuff there. Each one of these environments have their own API definitions, documents, structures. They all potentially vary in their quality and capabilities. They all may be very effective, and they may be deficient. So API management -- meaning, the way in which systems and components talk to each other -- the way clouds talk to clouds, the way containers talk to containers, the way microservices talk to microservices, the way platforms talk to platforms -- all needs to be thought of strategically. Because without a thorough understanding of



API performance and management, things can go awry very quickly. We can expose a service to the world, and all of a sudden, without the proper API controls, we crash our whole environment, because we didn't think through its implications.

(SLIDE)

So API management needs to be pursued in a different light. It needs to be pursued in the same light as our applications are -- meaning, APIs are assets; they're code, just like applications. We've got very sophisticated application management environments; we've got very sophisticated application development strategies and capabilities; and until API management vendors came into the market, APIs didn't have that same benefit from those same disciplines -- that same rigor of management. Now, life cycle management is a high priority. For APIs, it needs to be as competent and capable as our life cycle management for our applications and for all other IT assets that we use. And in my view, it's best when integrated with an environment to expose various services to one another -- to exchange data (inaudible) with one another. That's why, as I mentioned in the opening of this discussion, so many platform vendors acquired API management vendors -- because that's an integral need for next-generation application development.

(SLIDE)

And so what's happening, too, is, the process and the application development stacks are kind of converging into these next-generation digital automation platforms, where you can develop and run a business process application.

(SLIDE)

So, too, are the various data and integration capabilities we need. They're converging into what I'm calling a hybrid integration platform. And that enables a great deal of connectivity and integration across hybrid IT, but it also allows data quality improvement and management when needed, because data quality across source and targets needs to be addressed so that they're usable. Without such, things can fail. At the heart of this, I believe, is a core enabler, and that's API management -- which includes a catalog capability, a gateway capability, and a portal for development and publishing subscriptions, et cetera.

(SLIDE)

And so when you look at all of this as a holistic system, digital automation platforms, hybrid integration platforms, and API lifecycle management are core enablers to digital transformation -- to next-generation digital applications, to link the services needed by the users to enable that journey that either the customer or the employee or the partner needs to go on.

(SLIDE)

So when you pull it all together, strategy meets integration and automation. How we determine our activities and linkages in our value chain; how quickly we sense, interpret, decide, and act against change, risk, or

opportunity; how we link the various experience platform operations and collaborative capabilities into a uniform platform, is done through this hybrid representation of a digital automation platform and a hybrid integration platform, linking to backend services securely with API management -- life cycle management as one of the core enablers of this. Because without it -- without API management at the core of these systems -- a lot of things can basically go wrong, in my opinion.

(SLIDE)

So to quickly kind of sum up. Rivalry and empowered customers are driving this digital transformation age, and application strategies need to respond accordingly. APIs now are the digital accelerants and digital enablers to backend systems. And as I said earlier, no IT system talks to any other IT system today without an API in the middle somewhere -- containers use them, clouds use them, microservices use them, and any next-generation internally developed capabilities in modern application development doesn't exist without an API at the heart of it. But they're also journey enablers -- they enable that journey, to link the people, the devices, the interface, the applications, the data, the process, the infrastructure -- all together. A [consistent APIs?] is needed to do that, and understanding APIs as a strategic asset within the context of a broader digital automation platform and hybrid integration platform is -- to kind of unite the business and IT teams and enable hybrid IT is a big deal. So APIs are a core enabler to this -- they are core to next-generation platforms. And I believe that vendors should be sought by enterprises that understand this and [that'll?] make API management integral to their digital application strategy.

(SLIDE)

So with that, thank you. And I'm going to pass it to Sumit for Progress, where he can share with you how Progress addresses these trends.

(SLIDE)

**Sumit Sarkar:**

All right. Thank you, Carl.

(SLIDE)

And so, yes, I'll pick up on some of these trends. And from Progress's perspective, just a quick overview. Progress is a global leader in AppDev for strategic business applications and digital experiences. And we're going to talk about -- I guess as Carl's airplane analogy started at 40,000 feet and landed on API, I guess we'll next go and pick up -- at the baggage claim, we'll pick up some apps and see how that all looks.

(SLIDE)

And so when we talk about -- with some of the trends and initiatives and digital strategies -- when we look at AppDev, I think one of the areas we want to talk about is planning for rapid delivery of new application

experiences in response to a lot of this digital transformation -- frameworks and platforms that we've talked about -- Carl talked about earlier.

(SLIDE)

And what we're seeing is that through that role of API management and app strategy, the API life cycle management -- it's really a core digital enabler and accelerator. And maybe your organization has delivered an API strategy, and so you might be asking, When are all these apps going to start flowing, and when are we going to gain our competitive advantage through increased app delivery? And one of the challenges is, an API strategy is part of the way there, but it's not the same as an app strategy itself. And we all know that [the?] expectations are increasing on enterprises and IT -- and I think even at Progress IT, we certainly are seeing accelerated delivery requirements. And so this is not something new, but it is definitely being accelerated by this digital transformation [and?] API life cycles that are providing more access to more data to more people.

(SLIDE)

And one of the challenges that's slowing productivity is, a lot of existing teams and systems just are not ready. And so again, we talked about the expectations increasing on IT. And we hear this time and time again from IT decision-makers. And then we also -- we've seen that with some of the platforms and architectures that -- when you have monolithic architectures, they aren't as -- they may be inflexible. The scale required to support more modern applications -- maybe like mobile applications -- are 4x, and this increases the request on the hybrid IT infrastructure. And then when we look at changes to delivery, we're having to release more often, sometimes within -- you know, daily, in some cases. And to support agile delivery, there's a lot of adjustments that have to happen to existing processes -- across maybe DevOps, other infrastructure teams, and so on. And so the waterfall methods -- I think Carl mentioned -- are slow. And these are all areas that -- you know, we're ready to take off and start to build apps faster, and we're not quite there yet, despite having an API life cycle management strategy in place.

(SLIDE)

And so when we look at this, we call this the enterprise experience gap. And so on the left, you have a lot of different experiences. And [we?] may start by building web apps and mobile apps; then you get wearables and chat bots and all these new, exciting experiences. And on the right you have the enterprise systems that are -- you know, one of them may be an API management system; you may have multiple identity providers; you may have other SaaS applications and other systems of records. But all this is -- are the core foundation of the data that your experiences will need to interact with to provide modern experiences on top of your data. And I think Carl mentioned that a lot of the apps were built by IT for employees, and so that's why we're seeing this -- the systems and teams just aren't ready for this shift. And a lot of these systems need to be refactored in some way.

(SLIDE)

And when we think about how to speed things up, we talk about the abstraction objectives. And so when

you start with apps and we're trying to deliver new digital experiences -- like, client libraries that interact with backend service -- that's the abstraction layer for developers, not so much the API. And then when you start with data, the API management is really the abstraction for the enterprise systems of record. And so there's different abstraction objectives. And then when you think about it from the apps perspective -- an app developer -- that library or SDK to work with backend services -- really, the API that they care about -- not so much the low-level REST or SOAP API.

(SLIDE)

So there are different objectives between application services and API management. And then when we look at some of the services that work in conjunction with API management strategies, these are the backend services and SDKs I was talking about. So we talk about client libraries. What this really means is, they're low-code SDKs or libraries that build common features -- things like authentication management, data parsing, encryption, offline/online caching, things like that that are common to apps. And SDKs and libraries allow you to build that with low-code approaches in a consistent way. And for faster app delivery, if you deliver an SDK to a developer in the language or framework they are developing in, whether it's iOS, Android, NativeScript, or Angular, that's what they're expecting to be more productive versus an API. Other services are for data access; authentication, if you have multiple identity systems beyond your API management layer; encryption protocols, if you need to start looking for device encryption beyond the service tier; things like push notifications, other engagement services and access to third-party cloud services, whether it's something like Slack or Salesforce or another system of record in the cloud -- you may want to access other systems; and then analytics from an app perspective and tracking some of the versions of your apps and some of the life cycle management around that. And these are all services and [SDAs?] that are delivered through application backends. And Progress Kinvey is our cloud backend. And part of the reason for partnering with Carl on this topic is because a lot of our customers are using our Progress Kinvey with an API management system, whether it's MuleSoft or Apigee or something -- one of IBM's systems. And so we've noticed this trend, and that -- for application delivery -- that Progress Kinvey backend services and SDKs are a key part of that for many of our customers.

(SLIDE)

And when we start to look at -- my slide's messed up here -- let's see if this will -- [advance?]. All right, so pretend those circles make sense and they go around the persons or the boxes. And what we look at here with high productivity for enterprise dev teams is that when you combine the app backend services and API management, you really get more agile and lean teams that can deliver applications faster. And so we hear from customers that they can deliver enterprise apps, whether it's web, mobile, or other experiences, 60 percent faster than with the traditional approach of building the integration tier, having a separate team for web mobile, having cloud and DevOps. And so when you have a hosted backend and SDKs that are cross-platform for developers, so that my developer can build web and native iOS and Android apps, you get much more agile teams and you can deliver a lot faster through those increased demands on your IT. And with these diagrams, we're really thinking about high productivity in terms of output for a developer or a resource and in IT or [on your?] delivery team. And really, this is increasing -- it increases the delivery of the app and the competitive advantage that [we?] talked about a little bit earlier -- Carl mentioned from that

book -- I think it was Michael Porter. But yeah, those are all areas that lead up to meeting those expectations that have been now realized through this first step of, [let's say?], the API management life cycle and digital transformation from that perspective.

(SLIDE)

And so when we move on from, let's say, the high productivity -- meeting those increased expectations, let's start to think about more contrasting and understanding the backend requirements for app delivery beyond API management. And so [I'll?] just talk a little bit more about why we see so many customers looking for [these?] cloud services and SDKs.

(SLIDE)

And what we see is that web and mobile app abstraction -- what we're providing is, we're providing abstraction for API management. And I know you might be thinking, okay, so we need [to?] abstraction for another abstraction -- yeah, that sounds kind of -- sounds like a lot. And it's -- well, it does sound like more abstraction (inaudible) abstraction, it's really -- Progress [has?] a new digital architecture.

(SLIDE)

And this is a four-tier digital engagement architecture for -- it's [coined?] by Forrester Research in response to some web and mobile -- so for the transition from web architectures to mobile. And it's not that different from what Carl showed earlier -- it's just rotated clockwise. And really, this -- let's say, this abstraction for abstraction architecture -- [you know?], leading firms have embraced this -- from, like, Netflix and Salesforce and some of our enterprise customers -- [especially?] Bell and Howell, Canopy Health, or Schneider Electric. And at a high level, this client tier is at presentation layer. It really separates the native, web, and mobile development concerns, as well as things like wearables and IoT from the backend services. The delivery tier really is responsible for delivering contextually appropriate content to their client tier. And from the aggregation tier, [it's?] providing discoverability and translation between app requests and the services tier. And the services tier is comprised of the API layer as well as the foundation systems of records and other services. So this is the architecture.

(SLIDE)

And when we translate that into the architecture for the Progress Kinvey services [in?] the additional tier, it's -- again, I mentioned, it's not that different from what Carl showed. I put a screenshot of the initial slide there for reference. This is a lot of exciting marketecture on this one slide, but really we're [saying for?] Progress Kinvey SDKs and services [designed for?] API management, they digitize some of those services along with other enterprise systems and authentication. And the goal of the data has to be contextual for the app. And you may have mashups of data from one or more APIs or data sources -- maybe you need to integrate to Workday or some other system that's not wired into API management. There's a lot of different APIs on the Waldo slide Carl showed. You may have custom queries that require new business logic to deliver that context -- the right data at the right time [on?] the right device. You may have data that is contextual just

for the app, beyond something that you can store on the backend of an API layer -- like files and preferences and things like that. And again, new engagement services -- notifications and other analytics. These are all areas that Progress Kinvey's SDKs and backend provide to applications. And when we break that down from the initial tiers we talked about, the client tier -- the SDKs provided by Kinvey -- I don't know if you can read that -- on the top in blue you have the applications -- [so?] you have frameworks at the top for developers and their SDKs. You can deliver a host of web -- progressive web apps, native mobile apps. And they interact with these SDKs and libraries. And they are really that abstraction that we talked about earlier, from the backend. They include things like offline access, data synchronization when you come back online, those engagement services like push notifications, location, and things like that. And that's really the client tier in this architecture. And then when we look at the delivery tier, we provide -- Kinvey provides business logic, includes the collection hooks to process data from those underlying services -- and so it can be customized for delivery to your client -- [let's say?] your apps. And then the business logic itself is written in JavaScript, so it's a language familiar with most client developers, and they don't have to rely on maybe the backends that are evolving at a different pace than the application requirements. And so that separation at the aggregation tier and delivery tier are important. And then from the aggregation tier, specifically -- and again, this provides no- and low-code integration services for data and identity integrations. And those really provide the bridge between those enterprise systems of record and third-party services and the services tier and the delivery tier. And then the services tier -- you know, we [said?] it was the API management layer. And having this -- API management is one tier, and then -- but you have other systems -- you may authenticate with other -- maybe you have a LDAP or some other system of rec-- identity management provider you need to authenticate with; you may need to access other source systems at a different pace as requirements come in and you're developing [a new release?]. And so again, you have those two different speeds of innovation between services, and decoupling these allow those frontend teams to be more agile and deliver applications faster.

(SLIDE)

And this is just a summary of the cloud services provided by the Progress Kinvey backend. We kind of went over these in the previous slide, but just quickly to mention a couple of ones that -- a lot of the organization that is building on top of API management really look for the end-to-end encryption and for compliance purposes. Device encryption -- that's important through this -- having the backend and client SDKs and libraries. Having business logic is -- microservices to build more cloud native architectures on top of existing systems. And then the contextual data storage and things like that are key for the contextual experiences. Then the -- I think the data synchronization and offline capabilities are really important, because the API strategies may require you to -- the devices and apps to be connected, and so this provides data management for use cases where you may be offline -- very popular in field services or any kind of field organization. And then there's other functions and features in here that -- like, live service to provide real-time updates to devices -- that just are part of the app experience and not really engineered for the API layer.

And then we wrap up with a customer example, just to give you a real use case of an organization. Canopy Health -- to leverage a lot of the -- Progress Kinvey and those services. And they had a API management layer built on IBM. And to deliver these web and mobile applications, they really needed these services, and they had to authenticate with a different provider -- I believe it was Salesforce -- and then deliver all

these services for their mobile and web applications to the standard backend and services. And so this is an example of an organization that had an API layer, and then the next step to deliver and realize their web and mobile apps, they turned to Progress Kinvey [in?] the backend services and SDKs. And if you want to learn more about this -- the Progress Kinvey is on [progress.com/kinvey](http://progress.com/kinvey). And really, we're looking to engage with more organizations that have adopted API management strategy and see if we can increase the productivity and [in-app?] delivery expectations you're going to face now that you've launched this within your organization. And (inaudible) -- that's a wrap-up for -- I'm sorry, I just forgot to advance the slide here -- the --

(SLIDE)

Canopy Health use case.

(SLIDE)

And then wrap up for questions here. So [then?] turn it back to 451 to moderate those.

**Colleen Milton:**

Thanks, Sumit. It's now time for our Q&A session. As a reminder, simply type in your question in the box below, and we will get to as many as we can. First question is: we use a dedicated API management vendor. How do I link it with my AppDev tool chain?

**Carl Lehmann:**

Yeah. Hi, this is Carl Lehmann. I'll take that. So a lot of API management vendors have been subsumed by other development platforms, as I mentioned. So if they haven't already begun to integ-- if the vendor that you have is one of those and you're not satisfied with the rate at which the vendor is absorbing it into a broader platform, what a lot of companies are doing is kind of relying on iPaaS vendors to become an intermediary between the API development world and any of the backend systems that they need -- at least that's from my perspective. Sumit, you may have a perspective on that. (pause) I think you're on mute, Sumit.

**Sumit Sarkar:**

Oh. (laughs) Sorry. Can you hear me? Yeah. You can. Okay. Yeah. Let me see the question. I'm sorry, can you repeat the question one more time?

**Colleen Milton:**

Absolutely. We use a dedicated API management vendor. How do I link it with my AppDev tool chain?

**Sumit Sarkar:**

Oh, okay. Yeah, it's -- so, back to the -- some of the things we talked about earlier in terms of application development. We talked about those client libraries and different -- if you have different development teams building in different languages, from the perspective of the backend services and SDKs, the SDKs are the way to provide the [abstraction from?] different AppDev frameworks and [IDs?] to access those services with low-code approaches.

**Colleen Milton:**

Thank you. Next question: I'm a developer. Why do I need to know about a value chain and an adaptive loop?

**Carl Lehmann:**

Oh, okay. So my intention there was to expose technologists to a strategic thinking practiced at business managerial levels. When you understand some of the process and cross-functional business challenges of an enterprise and the means by which they try to respond to rivalry or customer expectations, you have a better way of serving the -- your enterprise -- to serving your business, serving your team. It's good to be a technology master, but if you master business technology, you become much more valuable to your company.

**Colleen Milton:**

Great. Okay, next question: I feel like we are going API-crazy. Is it best to build an API internally, or use a vendor-provided API? My view is, you should only build an API if it's a last resort or use vendor API first. Maybe we need an API team that just develops APIs and supports the repositories?

**Carl Lehmann:**

Yeah, I'll take that, and then Sumit will probably want to add to it. So you're right, there is API-crazy out there. As I mentioned earlier, everything is enabled to be an API. And I agree with you -- it's best to use existing APIs as opposed to building them yourself. The problem with that is, the quality of APIs varies greatly depending on who develops it and what tool they developed it in. And a lot of APIs are hand-coded, poorly documented, with weak or nonexistent SDKs, poor examples -- and have been untested. Many APIs do not go through a testing part of the life cycle management -- cycle. So you really have to hope that the vendor API that you choose is of sufficient quality so that you can rapidly use it and rely upon its capabilities to do what you intend. In its absence, you may have to create some workarounds. And [an?] API team is something that is being practiced in many enterprises. In fact, many enterprises are creating API product managers. Those are folks that are developing APIs to publish and to be consumed by third parties. Those APIs that you consume yourself from third parties need to go through a quality control process, I think, before deciding whether or not they're sufficient or insufficient for the purposes that you need. And support repository -- yeah, most API life cycle management platforms -- or any strategy -- enables an API catalog for discovery and reuse once they're vetted, tested, approved, and put into production.



## Sumit Sarkar:

So I can also talk to that a little bit further from an AppDev perspective -- and maybe some of the things we've seen in terms of organizations who have a similar question. So we talked about when developers are building applications on top of some of these services, they use an SDK to abstract it against these app backend services. And so our backend has -- it also has connectors for this purpose, because those developers aren't looking to necessarily build an API. And so we have -- we therefore deliver -- [it just converges?] some of the things that -- the architectures between hybrid integration and some of the application development side. But we have rapid connectors for APIs and then we have connectors to databases. We have connectors to other types of systems that can be developed through [a?] flux integration code. And so these are the reasons we did [that?]. You don't have to become a -- if you're from a development perspective, if you don't want to go API-crazy, just -- you can build through backend services and expose it through a common service, and then the SDKs can handle that and abstract some of those away. So whether you're connected to an API or a source system directly or whatever the data source is, that's the kind of architecture we've taken -- [it's?] not to just only support APIs, but support a breadth of data sources, so you don't have to necessarily build an API to wire things up. And I guess one other area to talk about in terms of just API development -- if you do have to build an API, we certainly endorse the OData standard, which is a -- it's a specification for building consuming REST APIs. And so if you [had?] OData (inaudible), you can learn more about that if you do choose to build one -- and try not to go too API-crazy.

## Colleen Milton:

All right. Thank you so much. That concludes today's webcast. As a reminder, the on-demand replay will be available on BrightTALK soon. On behalf of Progress and 451 Research, thank you so much for attending, and have a great day.

END OF AUDIO FILE



# VIEW WEBINAR

### 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](http://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](http://www.progress.com)

Find us on [facebook.com/progresssw](https://facebook.com/progresssw)

[twitter.com/progresssw](https://twitter.com/progresssw)

[youtube.com/progresssw](https://youtube.com/progresssw)

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

[www.progress.com/worldwide](http://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 | RITM0025037

