Deloitte.

For Cloud Professionals
July 2019

# For Cloud Professionals, part of the On Cloud Podcast

### David Linthicum, Managing Director, Chief Cloud Strategy Officer, Deloitte Consulting LLP

| | |
|---|---|
| **Title:** | **Taming cloud complexity chaos** |
| **Description**: | As companies progress further in their cloud maturity journey, their cloud architectures are becoming more complex, and that complexity can sometimes lead to negative returns from cloud deployment models. However, the tremendous cloud value proposition—and the strategic importance of cloud computing in business—makes it worthwhile to utilize cloud computing best practices and governance principles to effectively manage complexity and obtain higher returns on cloud investments. In this episode, David Linthicum sits down with Bala Rajaraman to discuss leading practices in cloud complexity management and ways companies can use them to leverage cloud computing for increased business value. |
| **Duration:** | **00:23:07** |

**Operator:**
The views, thoughts and opinions expressed by speakers or guests on this podcast belong solely to them, and do not necessarily reflect those of the hosts, the moderators, or Deloitte. Welcome to On Cloud, the podcast for cloud professionals, where we break down the state of cloud computing today and how you can unleash the power of cloud for your enterprise. Now here is your host David Linthicum.

**David Linthicum:**
So, welcome back to the On Cloud podcast, your one place to find out how to make your cloud computing work for your enterprise. This is an objective discussion with industry thought leaders who provide their own unique perspective around the pragmatic use of cloud-based technology. Today on our show is Bala Rajaraman, and he’s focusing on the same stuff I am, and I'm sure we've been in meetings together over the last couple of years, but it's great to have him on the podcast and pick his brain. So, tell us what you do Bala.

**Bala Rajaraman:**

I work in cloud. I've worked on a span of cloud technologies, from public cloud to private cloud to hybrid clouds, applications in the cloud. So, I mean, I've been working on cloud for a long time across various dimensions.

**David Linthicum:**
So, I'm seeing a couple of trends in the business right now. I'm seeing that people are layering microarchitectures on top of microarchitectures without a typical understanding where they're generally going. And, so, they're doing five or six systems at a time and moving them into a cloud and building net new and doing some migrations. They may migrate a dozen systems, leveraging a certain public cloud, database standard, security standard, and doing it again with a different public cloud and different enabling technologies for it. And, so, we have a tendency – they would look at that as using best of breed and I can understand why we're doing that, but this is also going to create the complexity issue going forward. In other words, we're going to have issues going forward where the complexity is going to hit a tipping point, where we're actually going to experience, I think, negative value in the cloud. And I think that's going to be a huge tragedy, especially for enterprises that have invested a lot in the cloud going forward. I'm not sure we can change behaviors. I'm not sure we can get people to get into architectural discipline at the last minute, looking at common systems, things like that, and there are reasons why they shouldn't even do that. However, we're going to have to be dealing with the complexity issue no matter what. So, what's your take on that? Are you seeing the same thing in the marketplace?

**Bala Rajaraman:**
Yeah absolutely. I think some of the initial journeys to the cloud have been very siloed, or they have been experimental in nature. I think there's not been strict architectural governance, because sometimes governance is viewed as the antithesis of agility, which I completely disagree with. But I think what we need is a couple of things. There's certainly the need for experimentation. There will certainly be the name to exploit the best of breed services in different clouds. But at the same time to be truly agile from an enterprise perspective, you have to be able to blend the agility of building with the agility of managing. And, some of the magic elements here is that our – what degree of automation you need. If you don't build that degree of automation, it's going to be very hard to shift some of the traditional Ops functions into the Dev side of it to get that degree of agility. But if you don't build with certain disciplines it becomes almost impossible to automate and manage.

So, I think to me what is going to be really important is how you actually bring both disciplines together, scale it across the enterprise, and extend it to be a little bit more hybrid. And by hybrid I don't mean locations, but in terms of the different kinds of workloads, whether it is microservices and serverless, whether it is more traditional stateful workloads, and also integration to existing workloads. Getting that framework together and being able to address both the agility from a buildout perspective but also agility from a management perspective driven through automation. That discipline is going to be really important. Without that discipline it's going to descend into chaos, much like you mentioned, in that it'll reduce the – both the velocity of movement to the cloud but also the value that can be extracted from it.

**David Linthicum:**
Yeah, ultimately, we even gave it a name. It's got to have a legitimate TLA, three-letter acronym, and we call it cloud complexity management, or the ability to, in essence, layer a series of approaches, and also technology stacks, on top of it. And I guess we can have – run the danger of actually making it more complex. But the ability to abstract and automate ourselves away from the complexity and the ability to look at these systems as certain abstract things such as cloud infrastructure, storage compute, databases, things like that – security infrastructure. Even though there's different physical instantiations and different standards that we're dealing with, the ability to look at it at a much higher level and then decompose those down into the actual technologies as needed is going to allow people who are managing and operating these things to scale. And right now, I'm seeing, you know – kind of a core metric in the business is people are identifying that they have a complexity problem by the turnover with the CloudOps teams. And, so, these Ops teams have been – you know, have the unfortunate position of getting these very complex systems that they're asked to operate. These things are getting more complex. They're adding more databases and more storage systems and more cloud-based systems, and obviously they have a limited amount of money and a limited number of skill sets that they're able to do, and they really can't scale. And, so, we're starting to see the turnover occurring with the CloudOps and that really comes a primary metric that we're going to have, this issue going forward.

So, you know, that's my higher-level take on it and there's lots of details behind it, you know, putting things in domains and the ability to understand what those domains are, and then mapping those into tool sets, things like that. There's complexity behind dealing with complexity of course. So, what's your approach? What would you recommend that enterprises take a look at first?

**Bala Rajaraman:**
Yeah, I think abstraction has to be meaningful in the sense that I think abstraction is going to be critical, to your point about addressing complexity. But I think abstraction that abstracts the meaningful details away, like performance or quality as a service, or other characteristics that distinguish one service from the other, they still have to be visible across the abstraction, maybe not the implementation details, but at least the policies, or the metadata about them. And, so, I think the abstractions will evolve. The other thing which I think is going to be really interesting, and this is foundational to some of the strategic decisions that enterprises are making, is to gain agility in one dimension you have to standardize and be prescriptive in other dimensions. When we look at the evolution of the internet, no one would say that the internet was not dynamic and agile. But, by the same token, if you look at it from the plumbing perspective, there were some very, very hardcore standards. Whether it's you're talking about TCP/IP, or you're talking about DNS, you're talking about HTTP, there were some very hard and prescriptive standards, which then allowed a whole bunch of innovation to happen, because at one level the complexity was managed, which lets it achieve a higher degree of automation and a higher degree of innovation.

So, I think the same thing is going to happen in the cloud space, which is there has to be a degree of prescriptiveness. We're seeing some of that evolution with things like Kubernetes and so on and so forth that provide some of the abstractions, but at the same time there is reasonable capability – and there's a lot to build here – in terms of exposing some of the variabilities that are necessary at the lower levels. I think the evolution of some of those capabilities are going to be critical for managing complexity. I'm very optimistic about it because I think we've made some – as an industry—certain movements towards Kubernetes, building an ecosystem around it in terms of things like policy management and application management, the ability to have more declarative ways of doing it, versus getting stuck in the procedural automation world. Some of those standardized elements and prescriptive elements are going to be very key to address the complexity challenge that you just posed.

**David Linthicum:**

So, ultimately if we're dealing with an enterprise that's looking at solving this issue, and I think they're not going to be able to move completely to a Kubernetes platform, and there are certain limitations that come with leveraging containers and the ability to leverage legacy systems. And, in fact, they're kind of great for some of the net new applications as we move things forward, but 95, 96 percent of the applications are going to be migrated to the cloud. And, in essence, with different enabling technologies and different target systems, but the complexity is going to come from kind of a hodgepodge of various systems that are migrated to the cloud. The existence of ERP in the cloud now, which is immensely popular, and I can understand why – I just wrote a blog on InfoWorld about that, your ability to save money and simplify your world, and it's just really compelling going forward.

So, we get into the pragmatic aspects of this, and I think that enterprises, enterprise IT specifically, are a little wary of us telling them to throw tools at the issue, or even new technology at the issue. And, so, this, ultimately, is not necessarily a manage by magazine approach, but this has to be something where we provide them with pragmatic views of what will actually work and what they can actually do using tools that are off the shelf today. So, again, we have to look at particular problem domain to really give them an accurate diagnosis and prescription for what they need to do, but ultimately what will your guidance be around looking for key technologies to leverage to make this a bit better?

**Bala Rajaraman:**
I mean, I start from an interesting perspective, because I think there has to be a goal in a lot of these transformations. And as you mentioned very appropriately, I think the whole point is agility, but they're also getting factors like the complexity of existing systems and there has to be meaningful movements from there to more standardized platforms – not necessarily containerized but more standardized, more automated, so that you can get to some degree of agility in those systems as well. But what is foundational to me from an agility perspective – it's just like we've been thinking about these DevOps or – well, it's called DevOps, but DevOps tool chains – is how do you go all the way from checking (Inaudible) code to building it and deploying it and automated testing and so on and so forth? The same discipline needs to start emerging from the Ops side. Even though we call it DevOps, for the most part the Ops part is shortchanged. There are disciplines like SRE and so on and so forth, but we haven't formalized that into a good thought process around what is the tool chain for operational agility. And more and more of what I'm seeing is that it is not about the individual tools; it is about how these tools together address the lifecycle, and that is going to be driven by some common understanding of the environment, all the way from the infrastructure to the application, so the role of metadata becomes really important.

So, what we are really getting back, and some of us old guys kind of know that, is truly actionable CMDB. If I look at any cloud, I think their success is driven by automation, which is driven implicitly by some notion of a CMDB and the metadata around it. I think that discipline is going to be incredibly critical for enterprises going forward, and not just how you model the data, or whether you have a repository called a configuration management database, but how do your tools integrate using that metadata to be completely automated? And let me give you an example. If you are building a new application and it needs to be – you need to have the flexibility to deploy it in different clouds with different configurations, with elasticity and so on and so forth, you have to have a view of where it is, how it is configured at this moment in time, what services are dependent on it, to be able to say anything about compliance. And, so, bringing the compliance world with the deployment world, and obviously with things like monitoring, logging, cost metering. That cannot be done in silos. That has to be integrated around a common set of metadata. To me that is the biggest transformation that needs to happen from an enterprise perspective in terms of some of the more meaningful, important workloads with high SLAs and so on and so forth. That is the change that the industry needs to move towards.

**David Linthicum:**
What are some of the emerging best practices that are coming right now? I guess we can, you know, get on a soapbox and tell people they shouldn't make these architectural mistakes in the first place and that there should be a generalization of vision in the ability to kind of get onto the same cloud platforms, the same database platforms, the ability to do security in the same way using different models. Yet we're moving in these very – you know, agile, which is good, but also chaotic directions that are going to basically cause the issue. So, if I'm talking to an enterprise right now and giving advice in terms of what they can do now in 2019 to start moving to either solving a complexity issue, or more often than not, preventing a complexity issue, what should they be thinking about? Should there be a chief complexity officer that's basically assigned to playing whack-a-mole with the complexity stuff and thinking through it? This becomes kind of a vision issue. This becomes kind of a CTO and CIO-level kind of initiative that needs to be pushed through.

**Bala Rajaraman:**
Agree completely, and I think to me this is the redefined role of an architect in an organization. And as we went through the discussion, I think the responsibility of the architect is significantly different, in the sense that you cannot distinguish application architecture, from platform architecture, from automation architecture, from operational architecture, from compliance architecture. There has to be an overarching set of capabilities, so the notion of a chief complexity officer or a chief standardization officer makes a lot of sense. But I think the concept you allude to is the challenge that we see with a lot of our enterprises, which is how do you arrive at a more complete view from Dev through Sec into Ops and get the right standards, platforms, tools, integrations associated with it that promotes agility but also enforces operational rigor? That challenge has been – and that challenge is further complicated by the fact that now you have hybrid environments, and nothing is pure. It's neither as simple as everything is microservices, nor that everything is legacy. You have to deal with this hybrid complexity. And what we are seeing is some of the areas of focus – is I think understanding and architecting your data architecture is going to be foundational, because applications follow data. And as you start segmenting data, picking different application architectures, how data is managed across these become really important. So, to me, data is number one.

I almost would start with not from a DevOps perspective, but an Ops Dev perspective, which is we have to lay out the foundation for how various domains, all the way from how do you deploy applications, to how do you secure applications, to how do you back up applications, you have a prescriptive path. Now obviously that'll get refined over time, but you have to have that path defined, and then get Dev teams onto that platform and have the rigor of using these services in a consistent way.

So, in dealing with a lot of the enterprises I think to me one of the key patterns that I go through is around a strangler pattern to support this evolution in a methodical and a thoughtful way, and a strangler pattern is – a critical element of a strangler pattern is an integration framework. You have to have an integration platform. That platform has to be agile so that you can actually distinguish what you're building that's new with what's old, along with the application is an API but also databases. And, so, those are the kind of key starting points and things that I would really focus on in terms of driving this transformation: your data architecture, your integration platforms, your management platforms coming together and being – and taking a prescriptive point of view.

**David Linthicum:**
So, let's talk about the data complexity specifically. You're addressing that right now, which I think is probably one of the easier issues to solve but also one that's probably more pervasive within the enterprises. In other words, your ability to remove complexity from the way in which you store and manage data is going to be kind of key to you getting to the complexity issue. So, are there tools that we should be looking at there? You mentioned data integration tools and there are certainly ones that – they're past based – I mean, in my past life I was CTO of three different data integration companies. But – and those things are helpful as well, but there's also different kind of patterns around leveraging integration and the ability to abstract information. So, we're dealing with virtual databases, virtual database management, your ability to, in essence, automate the migration of data between the systems.

How much should we kind of not necessarily mandate, but recommend that we take some of these poorly-designed databases that we've built out there, and instead of putting abstraction layers on top of them, or basically binding them together with loosely coupled integration layers, the ability to actually change them, or have this deep MDM system which is able to manage the way in which we deal with the metadata, it's able to make more meaningful use of the information, but ultimately we're getting into the changing of the physical databases and moving to different databases? We're (Inaudible) from relational object-based databases and all these things that people typically don't want to do, because their data is everything, and they view this as completely negatively disruptive and something that could kill their business.

**Bala Rajaraman:**
Yeah, I think that's a great question, because I think we have to look at the entirety of the data lifecycle. So, certainly what do we need to do tactically? How do you isolate things that can be worked on strategically? And what is your end state? So, many of the things that you mentioned, like do I have to virtualize data, do I have to federate data through a good metadata system, but still maintain some of the performance characteristics and cost characteristics becomes important. Choosing the right architecture for – and providing, to me, a little bit more of a prescriptive view of the different styles of data, whether it's object, whether it is graph databases, whether it is other forms of workload-optimized databases, it's still important not to just say, "Oh, it's no SQL." But, to me, as an enterprise perspective, especially when you're looking at hybrid environments, choosing a data architecture is one element, but managing the lifecycle of that entity through things like backup, performance, security, integration, compliance – all of those become dominant characteristics as well of making a data decision. So, understanding those and being prescriptive about that becomes important.

The one thing that I – to your earlier point about complexity, one of the things that's continued to vex me a little bit is – I guess there are two dimensions to it. One of them, as you split your application into more atomic chunks, or microservices, and if you stay pure to the architecture, then your data gets fragmented as well. In many of the systems that we have today, and especially as you get into more transactional semantics, it's hard to get your data architecture, that is both performant as well as (Inaudible), in a variety of ways. So, I think there's still some blending of more traditional data architectures with more, or newer, application architectures. But these are the kinds of decisions you have to go through when you start refactoring an existing state. We have dealt with many clients who are looking at things like, "Okay, I want to move – I want to redo my data architecture." I think looking at performance, looking at your backup strategy, looking at your application-consistent backup strategies, looking at the instrumentation, the automation around things like backup and so on and so forth that you need for compliance. Thinking through that I think is going to be a very important capability that we can bring to bear for our enterprise clients, is how do you handle all of these elements? What is the level of prescriptiveness in terms of a point of view we can bring? To me there are still a lot of unanswered questions in this space.

**David Linthicum:**
Yeah, I think so. I think it's a good area of research. I've been focusing on it for a while just because I think it's a green field. I don't think a lot of people have been thinking about this. I think a lot of enterprises probably don't see it coming, and they have recently. I've had a lot reach out to me with some rather disturbing things that are actually receiving, like I said, negative value in the cloud, because the Ops complexity is way out of date. So, I mean, keep coming back to this podcast, but also there's some stuff coming up that we're going to be externalizing around cloud complexity management and please take a look at that. So, anyway, if you enjoyed this podcast make sure to like and subscribe on iTunes or wherever you get your podcasts. Also check out my past episodes – our past episodes including On Cloud Broadcast hosted by my good friend Mike Kavis and his show Architecting the Cloud. Take a look at Mike's stuff. Mike wrote the book Architecting the Cloud as well. And if you'd like to learn more about Deloitte's cloud capabilities check out DeloitteCloudPodcast.com and we'll let you know where to go. And if you'd like to contact me directly you can reach me at dlinthicum@deloitte.com. So, where can we find you there on the web, Bala?

**Bala Rajaraman:**
You can find me on Twitter or LinkedIn, Bala Rajaraman. @balar is my Twitter handle. You certainly can reach out to me there.

**David Linthicum:**
So, anyway, until next time best of luck with your cloud projects. We'll talk to you guys again soon. Thank you very much.

**Bala Rajaraman:**
Thanks, Dave.

**Operator**:
Thank you for listening to On Cloud for Cloud Professionals with David Linthicum. Connect with David on Twitter and LinkedIn and visit the Deloitte On Cloud blog at www.deloitte.com/us/deloitte-on-cloud-blog. Be sure to rate and review the show on your favorite podcast app.

Visit the On Cloud library
www.deloitte.com/us/cloud-podcast

About Deloitte
-------------------------------

As used in this podcast, “Deloitte” means Deloitte Consulting LLP, a subsidiary of Deloitte LLP. Please see www.deloitte.com/us/about for a detailed description of our legal structure. Certain services may not be available to attest clients under the rules and regulations of public accounting.

Please see www.deloitte.com/about to learn more about our global network of member firms.