



## For Cloud Professionals, part of the On Cloud Podcast

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

**Title:** Want cloud success? It's in the strategy, not the tools

**Description:** There are myriad issues and technologies floating around the cloud space: cloud security, 5G, Anthos, SRE, blockchain, Kubernetes, cloud complexity, and countless others. Each technology or design philosophy has its place in a sound architectural framework for cloud computing. In this episode of the podcast, David Linthicum and Mike Kavis sit down together to share their opinions and answer some podcast-listener FAQs about new technologies and cloud computing best practices. In a wide-ranging discussion, they cover both technological and philosophical issues, but they distill all the tech speak into one clear mandate: tools themselves can never be a panacea to problems an organization may have. Instead, organizations must plan wisely, remove complexity, and develop strategies that help maximize the benefits of cloud computing and drive real business value.

**Disclaimer:** As referenced in this podcast, "Amazon" refers to AWS (Amazon Web Services) and "Google" refers to GCP (Google Cloud Platform).

**Duration:** 00:26:07

**Operator:**

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.

**David Linthicum:**

Hey, guys, welcome back to the On Cloud Podcast. Today we have a special show for you guys as I'm sitting here with my friend, colleague, and host of Architecting the Cloud, Mike Kavis. Mike, how long have we known each other?

**Mike Kavis:**

About as long as we've known cloud, so that must be back in the 2005, 2006, 2007 area.

**David Linthicum:**

I think we started working together in 2006, and I think we got to know each other like in the SOA days I think back in 2012, 2013, because you were running at the same conferences, things like that. I remember listening to you speak, and when I walked by the place where you were speaking, there were so many people trying to get into your session there was standing room only, and then it spilled out into the hallway. So, people were kind of looking over each other's trying to see what you were saying. This guy has something cool to say, man.

So, ultimately, we have a few topics to talk to you about today, and some of this stuff has been provided by you the users. So, we're happy to hear that because I'll tell you what, it's important that we talk about things that we like, but it's probably more important that we talk about things that you guys like as **listeners** of the podcast. And first topic would be sensitive personally identifiable information, SPII, as it pertains to cloud info sec, information security. And this kind of took me aback because I know PII in dealing with the medical community and things like that, but I'm not sure I've heard of SPII.

**David Linthicum:**

Well, let me go ahead and make an assumption. I think it's really the fact that we can tell specific personal things around people, beyond information that's typically stored. So, in other words, it's not just your medical condition, and any conditions that you may have, but the fact of the matter is it's who you live with, perhaps things that you made a mistake in the past at. They're not necessarily something that would drive a criminal record or something, but it's sensitive to you, and you're, in essence, going to want to be protecting that information or making sure that information doesn't get out. And I think that when we look at cloud and information security, we don't necessarily separate those pieces. We do separate PII because of the legal restriction of doing it, but not necessarily sensitive personally identifiable information. You know, ultimately, it seems like something that we really need to kind of put in its own category. What's your take on this?

**Mike Kavis:**

Yeah, it's social security number, biometric, passport, medical. It looks like the combination of PII and PHI, right? So, just everything that makes up the digital you, is kind of how I see this.

**David Linthicum:**

So, what information would you not share at a hospital that you would share in other places? I mean, is it your ability to get your financial information, health information, all of the above? Is it the ability to kind of combine everything in kind of a hybrid—massive stuff that we don't want people to know about?

**Mike Kavis:**

It's all of that, right? So, obviously you're going to have your financial information, but like my health information. I'm not sure I want the world to know what all my health issues are and those types of things. So, a lot of that stuff is pretty personal and private.

**David Linthicum:**

So, how should we protect it? Let's say it is personal and private information. We don't want it to get out. You know, what are ways in which we're doing it? Is it identity access management? Is it deep encryption services? Is it common encryption services that span different clouds? I'm finding that the vulnerabilities are typically occurring not, because we're not protecting information that's in a single, siloed system, but if you look at a hospital information system, for instance. It's sharing information with clinical systems, sharing information with billing systems, sharing information with discharge systems, sharing information with doctors' scheduling systems. And it's going to be very difficult to protect that information that's being shared with so many different systems, and they're all using different encryption algorithms. Some of them may not be using encryption, some at rest and some not at flight, and some in flight at not at rest. It just seems like if we're going to move this stuff into the cloud, we have to think into certain commonalities in terms of how we're going to protect this data.

**Mike Kavis:**

Yeah, and it's not just the systems that house the data themselves. As these hospitals are becoming more digital, more connected, those devices – you know, IOT-type devices, are all an entry point for someone to get in and get information. So, it's not just about protecting the data, but it's protecting everything that's connected to an IP address.

**David Linthicum:**

Yeah, I think this really kind of comes down to the discipline of information security in cloud. I think that those who are going to win at cloud info sec are going to be those who are able to plan and build an architecture, and we're not doing that. We're building cloud computing platforms – and I've seen this a ton of times in the last five years, where they're going through short sprints. They're doing five or six systems at a time, either migrating or building net new, and they're using whatever cloud technology they think is going to be the right best of breed technology choice at the time, including security technology, governance technology. They're doing it again the following year and doing it again the following six months and doing it again the following three months. And they end up having a big complex mess of different security systems that on their own in the particular sprint that they're working on may have made sense, but holistically, we're dealing with cloud-based systems that have to exchange information and do so in a secure way, so we can protect SPII and other information. We don't do it very well. So, what's the answer there?

**Mike Kavis:**

Well, there's no perfect answer but the companies that have been more successful than others – I won't say the companies that have been successful because I don't know if anyone's been successful keeping bad guys out. But they start with, let's build a platform, and they start with identifying the minimal set of requirements for the first couple apps, right? Because if you go out and try to build security for everything, two years later you may surface your head and say okay for an app to come. So, they usually start with a couple lower-risk apps and they build their foundation platform, whether that's leveraging a platform as a service, or building their own kind of PaaS on top of IAS. And then they build all the controls – you know, take all the policies that they have to adhere to. Everyone's got a core they have to adhere to regardless of the app. And then they say, okay, here's app number one. Well, app number one, this one has PII data. Okay, now let's go add the 100 policies that go with that, and they incrementally do that. But it takes some discipline that says, "Business

unit, you can't move to the cloud until these controls and policies are in place." And that's the battle, is the agility versus control, right? And some businesses can't wait, so they go do their own thing, and then there's exposure. So, it's a really tough challenge, but starting with something's better than, as you were saying, each team goes off and does their own thing, and then you have gaping holes everywhere.

**David Linthicum:**

Yeah, I like the term agility versus control, and I think that's kind of a key takeaway from this particular discussion point. So, moving on, we've got somebody who just basically talked about the three biggest things that are getting hits on the web these days: 5G, blockchain, and Anthos. I mean, we've both posted blogs, I've noticed, that anytime I mention 5G it just gets hit 20 times more than the other things, same with blockchain, and then of course same with Anthos, which is a product that Google just released. So, hitting 5G, here's kind of my take on it. I went and did a cloud talk years ago, and did so in kind of rural Australia. And they kind of laughed at me, because basically everything I told them and their access to cloud-based systems, you know, as we're talking about game-changing technologies and ways in which you can leverage it and things like that, but they didn't have access to broadband.

And, so, I felt kind of silly in the fact that they had no way of leveraging the cloud because, they had no way of communicating – at least the public cloud – no way of dealing with systems over the open internet because they didn't have broadband technology, which was down there. Some of them had DSL and satellite technology, but it was kind of few and far between. So, 5G has the promise of bringing 100 megabits per second, something you would typically get in your house with a cable modem up and down, and do so in a wireless way that can go to basically any part of the country. And this should be interesting because it's game-changing in a way that lots of people don't have access to broadband, easy access to broadband, inexpensive access to broadband, and this provides it. And, also, in providing them with access to broadband, it provides them with access to cloud-based resources. What's your take on this?

**Mike Kavis:**

Yeah, I mean, it seems like everything we talk about is disruptive, right? And here's another one, and I liken it to like when the cloud came, if you look at things like big data and analytics, because you had access to so much compute on demand, companies like in biometrics and things like that were able to do research that was never available to them before because of the compute power. And I think the same thing's going to happen with 5G. Because we're moving huge bandwidth restrictions, people are going to do things out on the edge and in places that – with a lot of compute power that they've never been able to do before. So, you're going to see a generation of apps and requirements we've never seen before, and of course we don't know how to run those things because we never have.

So, it's just going to be another one of these things that's going to follow the hype curve, where early adopters are going to go out there and create all kinds of exposures and issues. Everyone's going to say, "See? I told you so." But eventually it'll come to a maturing point and an ecosystem will build around it. But I think you're going to see applications like you've never seen before because now you have all this capability. I remember when I started in this field, you know, eight meg of memory was a lot, right? And now we're talking about high-speed internet out in places it was never available for. I just think the apps that are going to come out of this are things we haven't even thought of yet.

**David Linthicum:**

So, tell me about Anthos. You've taken the course. You've learned a lot about the technology. We both attended Google Next and saw the thing announced, and we looked at it in context of where everything else was going. What's your take on the technology so far?

**Mike Kavis:**

So, from a strategy standpoint I think it's an exciting announcement. Actually, we just published an article on my thoughts on it the other day, and you know, each cloud vendor – the big three, if you talk about the big three – now have kind of a hybrid solution. But I think the Google Anthos one is the only one that's hybrid for all cloud. So, in other words, the Google approach is you can run anywhere with any cloud. At least that's the dream. So, I think it's interesting; however, it's a beta product, right? There's a lot of gaps. They know that, right? They released it, and they have a roadmap. It's in proof of concept mode in my opinion now. It'll get there.

My real question is, is being cloud agnostic really the way forward, right? And we've probably talked about this a hundred times. There are use cases where it is, but I think the strategy where everything must be portable everywhere is probably not the best strategy. So, for example, if I have a cruise line, right, and I have to talk to boats that are in the middle of the ocean without connectivity, well, a portable multi – kind of multi-cloud touchpoint or even on prem makes a lot of sense, but not every app needs that. So, I keep coming back to what problem are we trying to solve? I think it solves a niche set of problems, but I still think the firms that just want everything to be portable all the time are kind of missing the boat on cloud.

**David Linthicum:**

Yeah, I think that's it, and I think there are no magical tools that are really kind of coming down the line. And I always get a little put in protective mode when people start throwing these things out as it's really kind of the next coming in terms of what things are going to save us. The reality is we're not necessarily going to get through this without a lot of deep analytics, without dealing with complexity issues, and without leveraging any number of tools to fix those issues. And so, I think going forward this tool is going to be important to the mix and basically solving the issue, but it's not necessarily going to be the tool that saves us. I think no tool saves us, and I don't think Google's actually saying the tool saves us.

**Mike Kavis:**

Right.

**David Linthicum:**

I think what we need to do is kind of put it into perspective of the fact that, you know, this thing's going to take us to a certain point, but ultimately if you're going to solve – and I've been looking at this complexity issue in a big way in the last few months. Ultimately if you're going to solve the complexity issue in dealing with multi-cloud environments and things like that, it really kind of comes down to planning. You can have a very complex environment, and if you put the right tooling and the right approaches and the right automation and the right abstractions on top of it, it doesn't necessarily have a negative effect. If you're not able to really kind of think through that and get to the solution fairly quickly it's going to kill you. And I think we're going to have a lot of people that are going to be at negative value in the cloud because they're not dealing correctly and interacting correctly with the complexity issue that cloud seems to be bringing. What's your take on this?

**Mike Kavis:**

I agree, and if – just look at Kubernetes for example. Kubernetes has exploded and everyone's jumping all over it, but that's a really – Kubernetes is pretty complex, right? When we start talking about at scale and being able to port workloads everywhere across all different kinds of networks and databases and stuff, that's pretty complex, and I see companies going down a rabbit hole rolling around in Kubernetes, which is one of the big points I like about Anthos, is that it kind of managed Kubernetes. And all the cloud vendors are having their version of managed Kubernetes. But too often people say, "I don't want to be locked in so I'm not using any of that." And talk about complexity – you know, running and managing huge clusters in Kubernetes is IT plumbing at its best, and most companies aren't in the business of IT plumbing. And that's just one area of complexity. Now they have these 50 other areas. And I think sometimes we're designing ourselves into being paralyzed.

**David Linthicum:**

I think you're right. So, when you say building your own Kubernetes, are they building their own distribution of the Kubernetes open source tree, or are they basically building a replica of Kubernetes within their enterprises, doing the same patterns, the solution patterns?

**Mike Kavis:**

Well, they're taking Kubernetes and they're kind of building their own pass with it, per se. And that's a lot of work, and there's a lot – it's kind of like Hadoop, right? You get Hadoop in the ecosystem – Hadoop, you got Pig, you got Hive, you've got this, you got that. All of a sudden there's 20 things. It's kind of similar in the world of cluster management. So, you know, companies are kind of rolling their own Kubernetes as a service, and I wouldn't recommend that. I think it's great to the end user, the developers to have that, but I don't think companies should be building that themselves. I think they should be leveraging solutions like an Anthos, like GKS, like – you know, each cloud vendor has their managed solution. That's a lot of work, right? And if you have a managed solution, it gets passed. It gets updated. One project we worked on a few years ago, we spent most of the time on the project standing up and running Kubernetes, more than writing the application. And then as soon as we launched the application, all of a sudden, you've got to patch a bunch of Kubernetes clusters, new software coming out. And it's a repeat cycle.

**David Linthicum:**

I always tell people OPC is good. OPC is other people's code.

**Mike Kavis:**

Yes.

**David Linthicum:**

You don't want to maintain things you don't have to maintain, and if you get it in the hands of the experts – that's why cloud computing is so brilliant, is that not necessarily we're consuming technology in a different, more economical way, which we are, but, ultimately, we're pushing the responsibility of maintaining and building and patching these systems and making them wonderful unto themselves, you know, the folks who are delivering this stuff on demand. And the great thing about that, it's also easy to switch. In other words, if something goes wrong, and we have to migrate over to another tool set, which is perfectly fine since it's on demand, we don't have to necessarily decouple something from an existing on-premise system. We can just point the system at something else. Of course, it's not that easy, but it's much easier than it was in the on-premise world.

So, something that may be in your wheelhouse is SRE, or site reliability engineering, versus production support engineering – pretty technical DevOps oriented question. Where do you fall on this?

**Mike Kavis:**

Yeah, so the answer is it's all good, no matter what you call it. [Laughter]

**David Linthicum:**

Whatever acronym you want to pick?

**Mike Kavis:**

Yeah, they're names, but the thing about SRE – so this is – for those who don't know, site reliability engineering was kind of Google's approach to production support. And the Google model was, you know, you build it, you run it, so this was developers trying to come up with a way to run ops. These are talented people who come up with a way to run what they build. And a typical SRE – and Google, at least initially when they wrote their papers on it, is they spent 50 percent of their time doing the monitoring login operations, but the other 50 percent of their time coding their way out of a job, basically automating all the things that they were doing manually. And they were also working side by side with the developers, so they're in alignment with the product or service they're building, as opposed to the traditional I build it, I throw it over the wall to this team who supports everyone, right? So, it was very focused on the product and SLAs and SLOs and all that.

I think it's a good thing for enterprises to adopt. I'll caveat that with depending on where they are on their DevOps journey, right? So, I've seen clients who don't even have automated CI/CD pipeline, and they're going, "We're going SRE." I would not advise that. The companies that are successful have already kind of solved the automation problems. They've solved the automated test problems. They've got consistent environments. They've broken down some of their silos so it's not a radical shift to shift more stuff left. So, I think if you're pursuing SRE, hopefully you have some of the basic fundamentals of DevOps in your place. Your culture is receptive to change and those types of things. That's where I see it work. Otherwise you can stick to more traditional methods that we've mentioned here.

**David Linthicum:**

Yeah. Are we overthinking this? That's kind of what keeps coming up, and I read a few articles to prep for this, and you know, ultimately, I kind of think, "We're really getting down into the same patterns that we're doing in each of these things, plus or minus 20 or 30 percent." And, ultimately, it's okay to reuse existing IP, problems that other people solved. But if we're spending too much time on this, on figuring out if it is SRE versus production support and engineering, I don't think we're necessarily solving the problem. We're just thinking through it too much. What do you think?

**Mike Kavis:**

I agree. I think you look at concepts like SRE and some of the other buzzwords, your takeaway should be, "What problem did they solve?" You shouldn't be trying to copy it, right? You look at it and say, "What did they solve?" Well they shift responsibility more, closer to the developer so the people who are solving problems are more familiar with the code? Well, that's a good concept. I like that. How does that make sense in my organization? It doesn't have to be SRE. You could do things that apply that concept, and that's really the message here. I've written about this a lot, about DevOps. DevOps is about eliminating bottlenecks and shifting stuff left. We started with the build process, CI, continuous integration. Then we went to the environment, CD, deploy consistent environments. Then we shifted testing left. Then we started DevSecOps, started shifting security left. Now people are shifting tier one, two, three support left, and operations. It's just a natural evolution of we keep moving – really the future of IT is IT's providing core services, and the business units are going to be building and running their stuff. That's where stuff's going. Whether you do it in an SRE model, you figure out how to do it within your company. That's all that really matters.

**David Linthicum:**

Yeah, I agree with that. So, last question: How big or small of a role does Kubernetes have in a hybrid world, and is it future-proof? So, we already kind of hit upon that, but you know, kind of my take on the Kubernetes thing, and I always thought Kubernetes was a cool technology when it popped out because we have these container-based systems. By the way, they didn't scale, and we didn't have scheduling. We didn't have core orchestration services in there, and then we had Mesos come along which is the Apache product, which is free, open source stuff. Lots of people were jumping on that bandwagon.

A couple of proprietary products that came from different container vendors, and then we had Kubernetes, which kind of popped out of Google as this core project. And when, you know, I kind of saw what was in this thing and did some write-ups on it and did some experimentation on it, I kind of got everything right. But, you know, here we are a few years later, and I think that everybody is just so enamored around building ecosystems around Kubernetes. They're in a sense pushing their entire cloud strategy, their entire product strategy if they're a product-based company, technology strategy if they're an enterprise technology company, toward Kubernetes. And that's a little scary to me. We have a tendency to kind of put all our eggs in one kind of cool, hype technology basket, and I'm not sure Kubernetes is really there for that purpose. What's your feeling on this?

**Mike Kavis:**

Well, first of all, nothing is future-proof, right? And everything is an evolution of the thing before, and it's no different from Kubernetes. And I'll tell you a little story about the evolution of containers, right? Well, many years ago when Docker first – before it was Docker it was a company called dotCloud, a PaaS company, and I was meeting with Solomon Hykes and he was showing me this concept – it wasn't called Docker yet, but you know, this is what we used to do live migrations. And long story short they open sourced Docker and it took off. Now Docker was the container run time. That's the only thing we talked about three, four years ago, right? And then all of a sudden, the orchestration engineers, and they were building Swarm, and Mesos was out there, and Kubernetes.

But under the cover was technology the PaaS solutions were using under the covers, so they could make changes while the developers on top are abstracted from it, right? So, now this stuff was open source. It was great. Developers can do this, and now we're off basically building PaaS. And what's really happening, and you see it with Anthos and some of these other things, is all that stuff's going back under the covers. So, I think Kubernetes is a key technology for the future, but I think in a few years it's going to go back under the covers. But it's going to be a force that's driving high availability and resiliency and clusters and all that stuff.

**David Linthicum:**

Yeah, and it's not a business strategy, nor is it –

**Mike Kavis:**

It's not.

**David Linthicum:**

--Or solution architecture, and it just keeps popping up. And the thing is, I've learned enough about being in this industry a ton of time – we had kind of the same thing with XML. People were putting – I was in the integration space, and people were putting XML as really kind of the second coming of standard base integration, to do XSLT to transformation and things like that. And I spent a lot of time and a lot of articles kind of pointing out that that's not necessarily a smart idea, for the reasons that are pretty obvious. This is a text-based system, and you know, pattern matching and things, these huge, huge performance issues.

But, it seems like if everybody seems to be technically correct, or like politically correct with the technology stuff around the Kubernetes stuff, and not necessarily looking at it in its pragmatic form, the ability to provide these capabilities is behind the scenes stuff. And, so, we leverage the core capability of what it's able to do. They're there to serve us. It's the ability to kind of orchestrate these containers and allow them to scale and, you know, doing clustering and things like that, but that's pretty much it.

And I think, ultimately, even though you have security-based ecosystems and database ecosystems, things like that that are going to help out, I think it's unhealthy if we focus too much on technology in essence, you know, providing us with a path to success. And I think it's not future-proof. We're going to be thinking about something else in three to four years. That's kind of the way we do things. And we'll be on to whatever, and Kubernetes is in essence going to be built into the infrastructure that we're leveraging.

**Mike Kavis:**

Yeah, I totally agree. And at the beginning of this we talked about SOA, and I remember working in a company, trying to push SOA and got nowhere. And, so, one day I pushed. "Here's your business problem and I'm going to solve it," and I never mentioned SOA or how, but SOA was the answer, and then we went and solved that problem. The same thing with Kubernetes – it may be solving some problems, but going around talking about Kubernetes isn't going to help anyone.

**David Linthicum:**

Yeah, a lot of mentioning of SOA and then a lot of mentioning of cloud, and I've learned quickly, like you did, to kind of remove it from the conversation and just talk about the solution patterns that we're offering. I think that's less polarizing. People get wrapped around the axle with the particular hyped,

buzzwordy technology. Of course, we talked about blockchain and 5G and all the other things that are coming down the line – machine learning and predictive analytics. But the reality is we have to look at how we're solving problems with stuff, and so people need to focus on that.

**Mike Kavis:**

Yeah, I just want to make one more comment on this. I think it's good that people rolled up their sleeves and figured out Kubernetes, because too often, you know, a lot of newer people come to the scene and everything's abstracted for them, and they don't really know how stuff works under the covers, and sometimes they build pretty crazy, bloated apps. So, I think it's good that people dove in and figured it out, but now we need to get over it and realize that this is plumbing and let someone else do that stuff unless there's a specific reason why you have to do it yourself. And let's move on to creating business value.

**David Linthicum:**

Absolutely, couldn't have said it better myself. That's a good way to end. So, that's it for today's episode of On Cloud. Thank you so much for listening, and if you want to hear more discussions between Mike and I, send Mike and I a Tweet at @DavidLinthicum – my last name is spelled L-I-N-T-H-I-C-U-M, and @MadGreek65, not geek, M-A-D-G-R-E-E-K-6-5, and let us know what topics that you would like to discuss on the show. And we'd be glad to give you some confusing answers but hopefully some things that are helpful.

**David Linthicum:**

All right, man. All right, so until next time, best of luck in building cloud computing architectures. We'll talk to you guys real soon. Take care. Bye-bye.

**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](http://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](http://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](http://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](http://www.deloitte.com/about) to learn more about our global network of member firms. Copyright © 2019 Deloitte Development LLC. All rights reserved.