Architecting the Cloud, part of the On Cloud Podcast

Mike Kavis, Managing Director, Deloitte Consulting LLP

Episode Title: **Edge or cloud? It depends on the need--and the desired solution.**

Description: Join host Mike Kavis and guest Simon Crosby of Swim.ai as they discuss the difference between edge computing and cloud computing--and when it's best to use each architecture philosophy. Mike and Simon cover edge computing, serverless, stateless vs. stateful computing, and how it's sometimes not only preferable, but critical to success, to choose one model over the other. Simon also gives his perspective on how to make use of the vast amount of computing power available at the edge to solve intractable problems, and on which use cases are suitable for cloud deployment and which require edge computing instead.

Duration: 0:23:19

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 Architecting the Cloud, part of the On Cloud Podcast, where we get real about Cloud Technology what works, what doesn't and why. Now here is your host Mike Kavis.

Mike Kavis:
Welcome to Deloitte's Architecting the Cloud Podcast. I'm your host Mike Kavis and I'm here with Simon Cosby, CTO of Swim.ai. Simon, welcome to the show – pleasure to have you. Tell us a little bit about your background the problems you guys are solving over there at Swim.

Simon Crosby:
Hey, thanks, Mike. It's a pleasure to be with you. You know, we're solving edge problems for massive amounts of data, massive amounts of data, and Swim is really an edge intelligence application run time that automatically builds and analyzes a live model of the real world, from data, so – and then automatically learns and delivers insights from that. So, the goal here is to bypass this complex process of having to get models built and to train them and then so on, to build the model from the data and just to learn on the fly and to do it statefully and to do it at the edge. And it's surprising how much you can do at the edge on quite a wimpy – you know, wimpy little computers and get profound insights in real time.

Mike Kavis:
I've been following what you've been writing the last few weeks and a couple of those topics I want to talk about today. One is, you know, there's a lot of buzz about serverless (inaudible) these days but when we talked about the edge, maybe serverless isn't the best fit there, and you've got a nice article about serverless anti-patterns, so let's talk about that for a bit.

Simon Crosby:
So, I don't want to come across as being negative about serverless, because serverless is terribly awesome for the cloud. And, in general, three key things have made the cloud wonderful. The first is REST, the second is essentially this notion of stateless computing, which is pretty much what serverless is, and then the third is databases. And that allows any old server to do my work for me, and that binding decision of which server is going to do the work right now, when the event happens, can be made late. The server loads my code, runs it, and hey ho, we're all good, right? It's just that the per-event latencies get to be very high. And so, whilst it gives you flexibility and saves a whole bunch of infrastructure concerns in the cloud because it can all be automated, it's not ideal for edge computing because often you know the location or locality of the thing that's giving you data. Data is arriving fast and furious and it's easy to process that in a stateful way, closer to where the data is being produced.

And so, let me give you a comparison. If I do some edge stateful processing of, say, traffic data from a streetlight, you know, I can (inaudible) memory and CPU speeds so let's just call that milliseconds at the edge, whereas to get to the cloud into a serverless process, then to a database and compute and then to database again, I'm looking at eight milliseconds. So, we're looking at, you know, seven, maybe eight orders of magnitude slowdown just to get stuff to the cloud to be processed, and I can do a lot with a billion CPU cycles at the edge. And I can do – and remember, I'm saving that every single event. So, stateful computation at the edge is really key, and stateful gives you this additional benefit, which is that you can then save cycles and do really cool things with it.

Mike Kavis:
Yeah, I wrote an article a few years back – I'm kind of abusing buzz words, but I said, "Forget big data. Small data is where IoT is at." And the example I used is out on the edge, I'm kind of seeing stuff stream in and seeing the what. So, the example I used there was a wind turbine and I was seeing shifts in atmospheric pressures, temperature, wind, and I was making decisions in real time on small devices, talking to actuators tell it to shift the blades. But at the same time, I was trickling data to the cloud to figure out why. Why does wind change do X and Y? So, I could get smart and adjust my instructions set out on the edge. So – and to your point – yeah, to your point the state and the real work's being done out there on the edge.

Simon Crosby:
Yeah, so you know, what you want to do is transform noisy data into durable insights, facts, predictions, which you can store as long as you want. Stick them – I don't care whether it's in the cloud or in a datacenter, but transformation into higher-level, more valuable insights is key.

Mike Kavis:
Yeah, and then another great post you wrote is "The Real World is Stateful," which kind of, you know, extending on the conversation we had, so talk about that post and where that's going.

Simon Crosby:
So stateless computing is a great attraction to cloud because it lets you do this binding activity of what server's going to do the work for my particular data item, and the database obviously is required. But if you're going to compute on the fly, there are some

important things you have to do, and I want to be clear about use cases here. I'll use traffic as an example. We do prediction on a per-intersection basis in many US cities, looking forward five minutes and predicting what the state of the intersection will be. And so, there's a huge amount of data we process, you know, something – the city of Las Vegas is 104 terabytes a day, okay? It's just a ton of data, something that you wouldn't want to even begin to think about paying for in the cloud.

Now, that's a huge amount of data, which is very noisy and not particularly useful. That is, you don't care that the light was green last Tuesday morning at 8:07. You really don't. The data has an ephemeral lifetime. What you want to do is digest the data in a stateful way, as quickly as you can, into something that is durable. And, in the case of traffic prediction, you want to know that that little bit of information was appropriately used to learn on the fly that certain traffic patterns predict other ones, right? And you just want to incorporate it in the learning. So, in the edge environment we have to be able to deliver the current insight at all times.

So, when I'm selling that insight to Google, or Lyft, or Apple, or Google, they need to be able to predict right now into the future, so I always have to have a current view and I have to be able to respond in real time. Now response in real – we're not controlling the lights, but we do other things, than just automation, where we want to stop a production run if we see an error, okay? So, you need to be able to respond right now if something is going awry, and that means that this batch mode of saving data on a disc and analyzing it later is just never going to help. And so, the key thing is taking the database off the application hot path.

So, this notion of REST, stateless, and databases, which made the cloud, is absolutely the wrong model for the edge where you want to do stateful computing, in memory, on the fly. If you want to save the data go right ahead; just don't do it on the application hot path. So, you need an in-memory model continually and statefully evolved, which can then be used as embedded into learning – so when I say learning, I mean self-training, reinforcement learning, and prediction and analysis.

Mike Kavis:
Yeah, and none of these concepts are new, right? I mean, just the technology's evolved. I remember when I first got out of college, I was working at a steel mill in the south and they had sensors picking up – you know, they had this thing had a hot strip mill which poured the steel and all that, and the sensors were there. They were collecting all this information. The difference was everything was being written to VSAM tape and a week later I'd get it, and a week later I'd produce a report, but these concepts of distributed and things out – you know, the edge back then wasn't so much hardware devices as they were, you know, edge devices or cloud, but there were things out there in the atmosphere collecting data. These concepts aren't new. I guess what's new is the speed at which we can process the real time nature and that type of stuff.

Simon Crosby:
Yeah, you're absolutely right. So, here's a crazy number for you on a little (inaudible) board. I can process traffic data at full rate from most of the Bay Area in about 19 milliseconds and do a prediction. That's about half the time it'll take a packet of data to get from a datacenter to the cloud. So, it's really important that we realize that there's a vast amount of unused compute at the edge, and that if we simply harvest those cycles, we can do amazing things. Now let me give you an example of costs, okay? If I solve the problem of traffic prediction for the city of Palo Alto in California, that's about $8,000.00 a month in AWA (inaudible). When we do it in prediction, it's no new hardware. We're just using spare activity-six cycles on the traffic management system that they already have.

We're learning and predicting for every single intersection in the city, presenting an API in Azure, which can be accessed by Uber and Waymo and Lyft, which gives forward predictions for every intersection, okay? No new hardware, so the marginal cost of producing the insight is zero, okay, versus tens of dollars per intersection per month. So, it's really, really important. You know, I see people failing at these big data/learning problems all the time, firstly because they think big data is it, so then they end up with batch (inaudible) insights, second because they don't know how to stand up these complex cloud data pipelines, third they don't have the data scientists to do the work, and fourth, it's just so darned expensive.

Mike Kavis:
Yeah, I think it's a case of a lot of people with hammers looking for nails instead of going to the toolbox and picking the right tools, you know?

Simon Crosby:
Well, I think most of the cloud-based learning pipelines and all the big data stuff came out of the cloud native world. So, you know, Spark and Hadoop and all these wonderful things came out of the native cloud folks, and they were using it in their way for their kinds of applications in (inaudible). It doesn't mean it applies to the real world and physical processes that we want to optimize at the edge.

Mike Kavis:

Yeah. I mean, I came out of a world where you had a datacenter talking to thousands and thousands of PCs out in grocery stores and pharmacies, you know, kind of edge devices back in the day, but it was the same thing. We had to write a lot of software that had very little processing power, and we didn't try to do that from corporate, right, because it was dial up back then. So, you had to do stuff out where the data was, so –

Simon Crosby:
Yeah.

Mike Kavis:
Yeah, it's like history repeating itself.

Simon Crosby:
Oh, and by the way, you know, you probably saw this, but Google is now out with their version of a Raspberry Pi with an attached Google TPU. I just can't wait to get my hands on one of those and use that for some real-world problems, because there is a massive shakeup coming in terms of our ability to learn on fast data at the edge.

Mike Kavis:
Well, that's pretty cool. So, let me ask you this. I keep seeing a lot of posts that says, "Cloud is dead. Edge is here." In my opinion you need both, but what's your opinion?

Simon Crosby:
Oh, absolutely you do. Look, the cloud – everybody in the world is just trying to get out of the business of being in IT, okay? That's what the cloud is all about, and it's fabulous for them, plus we have large enterprise applications being built cloud-native for the first time, and that's fabulous, okay? So, I'm all for it. I'm 100 percent in on the cloud. The key thing is not to buy off on cloud abstractions at the edge. So many of the tool sets that have been open-sourced, or the abstractions that we've learned, which served cloud computing, are just not the right ones to apply in an edge scenario. And so, I think what we have to do is have lightweight abstractions which solve edge problems and feed data into cloud-based applications in the right way for people to consume.

Mike Kavis:
Yeah, I couldn't agree more. Next question – I don't want to call it a fight but a stream going on about costs and stuff. And another great name – FinOps, like we need another something-ops, but they were talking about FinOps, the financials, right? And you had a great quote there. You say, "Most cloud revenue comes from keeping millions of copies in memory and occasional app execution." I thought that was pretty funny. Talk about that.

Simon Crosby:
Well, it's kind of true, right? I mean – and a senior executive at one of the large cloud companies admitted as much to me. You know, everybody's rushing to get out of the business of being in IT – for very good reason. It's extraordinarily expensive for enterprises to run all this stuff. And now it's all running, in a large part, in a duplicated fashion in the cloud. So, if you think about virtualization VMs, gosh, now I have lots and lots of copies of probably the same OS in memory all the time, and the applications run occasionally. But even if you go to my example of REST and Lambda, you know, you're lucky if you get one out of a billion CPU cycles actually doing some work, right?

So, think about this way. You know, my event shows up at some Lambda load balancer, then that – some server has to get back – it has to load my code from other storage, which is a REST interface away. Then my code eventually gets to run. I get to do a little bit of work, and then I wait while this thing goes and gets the old state from a database. And then I do a bit more work again and then I wait while it goes and puts a new value in the database, okay? And all of these things are a network hop away. So, I'm going to make this real for you. If you can walk at about a meter per second, okay, then one cycle at the edge is one meter, right? You know, it's the same as say – for me, and I'm just outside Seattle – to talk all the way to Portland and back, and then resume this conversation. Okay, it's like that, okay? So, we waste a vast amount of computational cycles, and because the abstractions help; they simplify it for humans. But we're wasting a ton of cycles.

Mike Kavis:
Yeah, it's a very interesting perspective and I can go through history of computing to you – the same thing. Like when I was on the mainframe, mainframe was just this big box with endless compute to the developer and he just wrote code, and half the time they were wasting stuff. But I'll go back to my retail experience. We had PCs with hardly anything on it in the store. The people who wrote

to those applications just had no memory, no storage, and they wrote differently than the people sitting back writing on the mainframe, or the big Unix box. And I think that's the same parallel we have today, writing in the cloud, versus writing on the edge.

Simon Crosby:
Yeah, yeah, that's right. Yeah, spot on. But you know what? I think in computing there isn't an awful lot that's really new. All we do is we chase the bottlenecks around, you know, and then we get to go and optimize them. So, we'll always have jobs, because we'll always have a new bottleneck, and we can go and optimize that one and chase it somewhere else.

Mike Kavis:
Well, plus we've always got to fix the crap we build, too, so that's –

Simon Crosby:
Exactly, yeah. It's a good place.

Mike Kavis:
So, this brings me to my last topic, and it's really about architecture, right? And what I wrote down here is there's so much innovation coming at us, right – edge computing, and we've got to embrace DevOps, and IoT, and big data, and machine learning, and AI. And with every one of these we tell our people, "This is a mind shift change. You have to change your thinking." And it's like, how many times do I have to change my thinking? At the end of the day, this is all ones and zeros, right? Is it – so my question is the great ones, the architects who've been through all this, is it really a mind shift each time, or are we getting too locked into the thing that's hot now, or what is the approach of an architect who's excelled through all these? What is their approach? Is this really a mind shift change each time, or is it just a way of thinking about systems?

Simon Crosby:
I think it's a way of thinking about systems. There are some big new things in the way that we can build now for clouds and the way that we are building. I think that, for example, even work practices and applications are changing dramatically. I mean, if you think about the last few years, the biggest change has been from a world of IT, to a world of DevOps, okay, where the people who write the application also run it, and run it at scale, and update it at scale, and partially update it at scale, and so on, whereas historically it used to be IT who was responsible for patching, right? And so, the learning of how to run things at scale is increasingly being built into toolsets which help automate the function for you.

But, you know, we are undergoing a radical change in computing from computing being something that you had to be deeply expert in all the layers of, to being something where you have to track the best way to deliver the applications to you and your community, given the current abstractions that you have. Nonetheless, I think that the marketing engines are going at full steam and full of BS, and here's one. Now they are saying that the future machine learning for the edge is one where you store all the data in the cloud. You then have somebody who's a data scientist who's going to create and train a model, and then they will deploy that to the edge on some device. Actually, all they want is your data and they couldn't care about the rest.

Mike Kavis:
Right.

Simon Crosby:
Is that right? Once you're addicted to their cloud, you're in. And so that's a fictional work practice for a fictional labor force which doesn't exist. It's just they don't know, and what they do know is they'd love to have your data, okay? And so, I think we're at a point where we have to call BS on certain things which are just fictional, right, and dig deeper. And it's hard for architects to know that, right? I mean, I have yet to find one serious industrial player who thinks they can go head to head against, say Google or Microsoft, in terms of machine learning skill set. In general, what you have are some good analysts, but they're normally running around a shop floor getting data from equipment. So, we have to call BS when the marketing engines get too far ahead of themselves, and I think this cloud cast – you know, this podcast is a really good way of helping to keep people informed of when they should ask the deeper questions.

Mike Kavis:
Calling BS is a full-time job these days.

Simon Crosby:

It is, yeah.

Mike Kavis:
All right, well, this has been a really cool conversation. I appreciate your time. Where can people find you and find these blogs and any presentations –

Simon Crosby:
So, the company is Swim.AI, and my blog is at Blog.Swim.AI, and I am @SimonCrosby on Twitter. Thanks so much for the opportunity.

Mike Kavis:
Oh, I enjoyed it. We'll have to do it again someday. So that's our show for today. You can find more podcasts by me and my colleague Dave Linthicum just by searching for Deloitte On Cloud Podcasts. We will see you next time on Architecting the Cloud. Simon, thanks a lot. I appreciate it. Sorry for all the false starts –

Simon Crosby:
Oh, no problem at all. Thanks so much for the opportunity. It was really great. And from a Deloitte perspective, you know, we're doing great things, so I'm really jazzed about that. We have – I didn't mention this, but we have – we're working on a couple of government contracts (audio ends).

Operator:
Thank you for listening to Architecting the Cloud, part of the On Cloud Podcast with Mike Kavis. Connect with Mike on Twitter, 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.

Copyright © 2019 Deloitte Development LLC. All rights reserved.