Deloitte.

The Deloitte On Cloud Podcast

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

Title: **Superclouds: A new solution for managing multi-cloud complexity**

Description: As multi-cloud has become the de facto architecture for many companies, complexity has emerged as one of the biggest hurdles to cloud value. Enter the supercloud, which is an architectural pattern that creates an abstraction automation layer, logically, on top of the multi-cloud environment. Superclouds provide a single control panel that spans an organization's cloud ecosystem. They can't hide poor architecture, but they can help manage multi-cloud complexity and provide more cloud value.

Duration: **00:19:29**

David Linthicum:
Welcome to this Deloitte On Cloud podcast knowledge short exploring a specific topic related to cloud computing. This is a short tutorial talking about the real-world concepts in the emerging world of cloud computing. I'm your host, David Linthicum, cloud computing subject matter expert, author, speaker, and managing director with Deloitte Consulting, and this is, "What's a supercloud?"

So, this is a topic that has been requested a lot in the last couple of months because it's risen up in the press, and it's something we've been talking about here on the podcast in terms of complexity and your ability to do multi-cloud management, all those sorts of things. So, figured it's time to do a knowledge short on what a supercloud is, what a metacloud is. They're all the same thing, by the way. And sometimes we're hearing it called a skycloud. Just get that out of the way. We don't care about buzzwords. No matter what you want to call it, it doesn't really matter as long as you kind of maintain the same concept, the ability to build this layer of abstraction automation that sits above the public cloud providers to reduce redundancy, make things simple, things like that. Let's get into what that's all about.

So, a few years ago, I pointed out in my InfoWorld blog that multi-cloud is really not about the public clouds it's built on. And we see the emergence of technology layers that sit above a collection of public clouds, and that's really kind of what multi-cloud is becoming. So, if you're looking at what a multi-cloud is, you have different cloud providers and in order to reduce the complexity, we're looking to create an abstraction automation layer on top of the clouds, logically on top of the clouds. They may physically run on different clouds, on-premise, things like that, which is able to operate cross-cloud services, things like security, operations, governance, FinOps, things like that, so we're reducing the amount of redundancy by replicating services that sit in the particular public cloud provider silos, pushing that up into a super layer called supercloud, metacloud, whatever, and you're able to solve the same problem multiple times in multiple cloud providers through a single layer of technology.

Security, instead of having different security layers that are native to particular cloud providers, we run one that goes across the clouds. Same with operations, same with governance, same with FinOps, even same with application development, even some application deployment layers based on Kubernetes. We get into federated Kubernetes where we can actually deploy layers that may only have an application that runs on a particular cloud provider but we're able to move it across logically between the cloud providers as we need to. A lot of that's not going on yet but we're definitely heading in that direction.

So, this supercloud, metacloud layer basically exists and lots of things are in it, including application development, operations, observability, security, governance, and other things that exist above a public cloud provider that are bundled together to make clouds operate and work together in a much more simplistic way. So, instead of 100 different services that are operating in a multi-cloud environment, we may be able to do it with 50 because we're combining some redundant services and pushing them up into this abstraction and automation layer which is the supercloud.

So, the terms beginning to emerge—supercloud, distributed cloud, metacloud—and that's my vote, by the way—abstract cloud, and some people are calling it skycloud. Even the term cloud-native is up for debate. Those are being defined different ways, so let's not get too much into the buzzword makers here, but to be fair to the buzzword makers, they all define the concept a bit differently, and I know the wrath of defining a buzzword differently than others do, as someone who's done it may times in my career, and, ultimately, the common patters of this, which is really what we're looking at, the common architecture of what a supercloud is, is really a collection of public clouds and sometimes edge-based systems, that are able to work together for a greater purpose by putting a layer of abstraction and automation above the existing cloud providers. And again, that's a logical layer. I mean, you physically run on different cloud providers even on-premises, but it's a set of cross-cloud services that's able to span the different cloud providers, even private clouds, edge computing, legacy systems would be included with that.

So, I think that this concept will be a single focus in the cloud world for the next five to ten years as we begin to put public clouds to work and we also move into multi-cloud, and then we understand that there's a certain amount of complexity that we have to work through if we're going to do that. So, if you're leveraging a multi-cloud system and lots of you are doing it, I think 95 percent in some of the surveys I've seen, you're going to have to maintain a layer of technology that sits above the different public cloud providers to ultimately manage those systems. We're managing them as one particular environment, not as four different clouds, five different clouds, things like that.

And by doing that, we're able to do it much more efficiently, and we're able to do so in kind of an operational state where we're able to use fewer resources, fewer skills to maintain the same collection of cloud providers within our multi-cloud and within our enterprise, which also may include legacy systems, edge-based systems, things like that. So, we want to leverage cloud providers through abstract interfaces to access specific services such as storage, compute, artificial intelligent data. We want to support a layer of cloud-spanning technology that allows us to use these services more effectively, and a supercloud or metacloud removes the complexity that multi-cloud brings these days. Also scaling operations to support multi-cloud where it, in previous layer, was too complex, too many skills were needed, therefore it wasn't cost-effective to operationalize those systems without some sort of a cross-cloud layer.

So, keep in mind the goal is we're trying to combine as many things as we can in this cross-cloud layer of services which they're calling a supercloud. So, while the goal is to have a single layer of security, governance, operations, even application development and deployment, this is really what multi-cloud needs to become to minimize or mediate the security issue that we're dealing with when we're leveraging multi-cloud. We just have too many services under management, we're trying to do too many things using whatever cloud-native tools that that particular public cloud provider provides, and this allows us to think differently in moving those things up into a layer where we move a lot of those redundancies. We normalize the services that are under management, therefore make it more simple to run.

So, we're just building right now more complexity that will end up making multi-cloud more of a liability than asset, and we're already seeing aspects of this, and people are looking at the cost of cloud computing, in terms of surveys are coming back that they're looking to even stop cloud production until they figure out the cost issue. When you look at it detailed, what's going on is they're hitting a complexity wall, they're just getting too many services under management that they can't operationalize at the current level of spending that they want to spend, and also there's not a lot of discipline that's going into the cloud cost management stuff the ability to implement FinOps programs, things like that to make things more optimized. So, we're fixing those things as we move forward, but this is a step in that direction. In other words, optimizing your cloud architecture by pushing as much redundancy as you can up into a single layer, we're able to solve a problem one time that's able to solve across different cloud providers.

So, one of the things that kept popping up when we were talking about kind of the emergence of the supercloud concept is lots of vendors out there were claiming that they were selling a supercloud or a metacloud already, and, so, that was a common question, but I need to point out a few things. Metacloud as it exists today is an architectural pattern, not a specific product or technology. Granted there's lots of confusion out there these days around different buzzwords and what concepts mean, and they redefine them basically versus the way they see their world, those sorts of things, but at the end of the day, it's an architectural pattern, it's not yet a particular single product that you can buy. It's made up of lots of different technologies in the stack, but unto itself it's not a single product.

So, today metacloud, supercloud is formed by any product, technology, or architecture standard that works across two or more public cloud providers. So, anything that address cross-cloud systems such as security, storage, networking, or whatever, although the product can certainly be a part of a metacloud or supercloud architecture. The product itself is not a metacloud or supercloud, at least not the concept that I presented in the writing, and what people are talking about there and kind of the aggregation of the opinions that are out there particularly in the marketplace. So, that's not going to serve them well if they try to sell an architectural pattern as being what their product is.

So, the other question we got is the metacloud primary service provider, and in other words, there has to be some major cloud service provider that's part of metacloud or supercloud. Most of the major cloud service providers hold a view that they need to be part of what a supercloud and metacloud is, and really, they're only half right. Keep in mind that a metacloud or whatever you want to name it is a layer of technology that logically operates on top of two or more public cloud providers. So, the key word there is logically. So, a metacloud or a supercloud dashboard could logically operate across using cross-cloud security services, cross-cloud operations, cross-cloud governance, cross-cloud common storage systems, et cetera. Anything that provides abstraction from the native or proprietary cloud services that only works on that particular cloud provider that sits above those providers that works across providers. In other words, it's not localized on a specific provider only providing services for that provider. It's able to span different public cloud providers, even edge computing system and legacy systems.

So, the metacloud architecture technology stack may physically reside, as we mentioned earlier, on any cloud platform. Has to run someplace. Either it's going to be on premise in the cloud, edge systems, whatever, and ultimately it may run and span different cloud providers even though it logically runs on a particular cloud provider. So, it should be no issue where it actually runs. You should think about this as kind of a logical architecture that sits above the specific providers but, like I said, it's got to run someplace. And, so, you are going to need a platform to deploy this stuff on. That's typically going to be a single platform, not multiple platforms, but it deploys on one platform, and it is able to operate across different cloud platforms using abstraction automation, as we mentioned earlier.

So, this is a bit confusing since we're looking for cloud-agnostic technology to drive the metacloud or the supercloud, but the stuff must run somewhere, and that's typically on a major cloud provider. And that's typically going to be my choice because it's cheaper to run on a cloud provider than it is to deploy a physical server which is going to be in your data center on premise. So, if you deploy much what drives a metacloud, meaning cross-cloud systems that mostly resides on a specific provider, then I agree you can consider your cloud provider as your primary metacloud service provider. But I'm not sure it makes much difference in the outcome since it really doesn't matter where it runs.

So, keep in mind that adding a layer above multicloud deployments or including legacy and edge computing—we can certainly include that as well—does not remove complexity. It actually hides it. Another comment, and possibly a correct one, that we got back on the feedback on what a supercloud and metacloud is, is that multi-cloud complexity is a result of selecting many different types of cloud services from different cloud providers. It typically happens when those charged with driving innovation prioritize best of breed cloud services that will more closely meet the project's business requirement over the impacts of those decisions will have on the complexity of the conglomerates of the systems that we're deploying. Complexity and heterogeneity are the results, and that's where the—we can't prioritize, we can't deal with logical architectures that way.

So, keep in mind complexity can be avoided when IT organizations restrict the use of all but a small number of approved cloud services. So, this does limit your developers and innovators. So, what we're saying is we have a complexity problem, and you can't use any service you want. You have to use the services that are of this particular cloud provider. So, the tradeoff there is we're going to limit their ability to leverage the right technology they need to be most innovative and provide the most innovative solutions. So, by doing that, we're going to restrict their ability to add value to the business because most businesses are valued, for the most part, on their ability to innovate and get new innovations into the marketplace. So, we always want them to have the best tools and technology, and that could be whatever best of breed technology it's going to take to make it happen. So, for complex systems already in place, hiding architecture complexity stemming from poorly designed systems is not a best practice. This kind of complexity is difficult from complexity created by onboarding cloud services that drive best of breed values. If complexity results from poor design, the business will be better off if you fix it rather than hide it.

So, what about using—or the question around using abstraction layers as causing latency? If you think about this, we're talking about metacloud as something that sits above different physical systems and uses these systems through an abstraction layer. So, we're dealing with many systems using APIs that are representing many different physical different things—physical storage, physical databases, physical compute processes, physical containers, things like that. So, this becomes kind of a cool question to answer because this is a frequent comment on any kind of abstraction thing, and if you think about it, if a metacloud, supercloud is an abstraction layer, then we're going to have to think about performance. So, ultimately, if you add another layer between the native interface and those who are ultimately consuming that service, there will be a delay in request and response time.

That's just a law of physics. In other words, if we're going through another layer, there's going to be a certain amount of latency or price that's going to be paid for us going through that layer. Kind of keep that in mind. For instance, say we use enterprise storage interface abstraction that provides a common interface for all public cloud native storage systems. The developers do not need to write each specific sometimes proprietary API for each public cloud provider because the interface writes to a single abstract API application programming interface. There's CLI, call level interface, that uses automation to deal with the differences. Yes, there was some latency tradeoffs because additional processes and IO need to occur, but in most cases, you won't notice it. It's going to be hidden because processes are faster these days, networks are faster these days, so it's going to be there but you're probably not going to notice. It's going to be not a big deal as far as building the systems.

Of course, for the metacloud, it's not going to be that simple. This is about the abstraction of storage, the compute instances that allow for easier and more portable ways to interact with many different types of cloud services that exist on many different types of public clouds. Most of what makes up a metacloud or a supercloud abstraction around the operational processes such as security, core operations, governance, metadata management, other things that would be overly complex and laborious that we had to deal with them using each native system in each cloud instead of working with one storage operations layer, which we're doing so through abstraction, you would have to deal with four or five, making sure you have the skills on the staff for each public cloud provider and local cloud services, and this becomes where the cost comes in. So, again, complexity equals costs.

If you're dealing with five different storage systems and using five different native interfaces and five different native dashboards, we have to keep five different skills around to make that happen versus us leveraging some sort of an abstraction layer that's able to operate across clouds where we do so through a single abstract API, a single layer of automation, single dashboard, things like that. So, we're not always going to be able to solve that problem with abstraction. I understand that this is an architectural pattern that's going to be applicable in some instances and not others, but just the fact we have this in our arsenal to reduce complexity and optimize some of the cloud deployments that we're moving out there, and certainly the complex ones that are around multi-cloud and integrating legacy systems and edge-based systems as well, so it's kind of key to this.

So, most of the questions that I get about metacloud, supercloud, certainly the stuff I've written and, some of the stuff I've talked about on the podcast were based on sound logic, I think. So, people are asking good questions for good reasons as to what this is and what value it's going to have. It’s important to question any new concept and weigh its pros and cons. That's core to what this is, and by the way, you can come back on this podcast and give me comments on what you think. This is an evolving process. We've been talking certainly within the firm about common services and cross-cloud services to deal with multicloud for the last three to four years, certainly the cloud complexity management stuff, things like that. At the end of the day, what you're building. You're mediating cloud complexity is a supercloud, a metacloud. You're creating abstraction automation to mediate the complexity issue.

However, we do know that it's going to be emerging over time, and it's going to be changing over time, and people are going to have input into what that is. And, so, it's going to be evolving technology, but it's really kind of easy to say that for the next probably ten years, we're going to focusing more on this than the particular cloud providers, while the cloud providers are going to be important. And by the way, they're still going to be making money and growing this kind of environment. As we deal with the challenges with complexity, as we deal with the challenges of other systems, then we're going to have to be able to figure out how to make this happen. It's a cool concept, it's out there now. We will inform you about the evolution of supercloud and metacloud on this podcast, but you guys keep that in mind.

Anyway, if you enjoyed this podcast, make sure to like us, rate us, and subscribe. You can also check out our past episodes, including those hosted by my good friend, Mike Kavis. Find out more at deloittecloudpodcast.com. If you'd like to contact me directly, email me at dlinthicum@deloitte.com. So, until next time, best of luck on your cloud journey. Everybody stay safe. Take care.

Operator:
This podcast is produced by Deloitte. The views and opinions expressed by podcast speakers and guests are solely their own and do not reflect the opinions of Deloitte. This podcast provides general information only and is not intended to constitute advice or services of any kind. For additional information about Deloitte, go to Deloitte.com/about.

This publication contains general information only and Deloitte is not, by means of this publication, rendering accounting, business, financial, investment, legal, tax, or other professional advice or services. This publication is not a substitute for such professional advice or services, nor should it be used as a basis for any decision or action that may affect your business. Before making any decision or taking any action that may affect your business, you should consult a qualified professional advisor.

Deloitte shall not be responsible for any loss sustained by any person who relies on this publication.

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