**NETEVENTS**

# EUROPEAN MEDIA SPOTLIGHT

# "INNOVATORS IN CLOUD, IOT, AI & SECURITY"

**DRAFT**

## *Reimagining Connectivity to Meet Fourth Revolution Needs: Applicable-driven Networking*

Guest Speaker Presentation by:

**Galeal Zino, Founder, NetFoundry**

**Galeal Zino**

Nice to be here today. Thank you for the invite to the NetEvents folks. Excellent to be here in this room full of experts. Looking forward to some good conversation throughout these panels, including here.

Wanted to speak specifically about what is called the Fourth Industrial Revolution and the role of programmable Internet in that revolution and I'll back up just a step with some old news. Not fake news or false news, but old news, and that's a software eats world concept and we all get software eats world and it makes sense and we understand it and it is everywhere. However, in many ways, it's actually the first chapter of a longer book. Maybe even the first book in a longer series of books. Software eats world is the first part of the enablement of a Fourth Industrial Revolution.

To think about that for a moment and think about software, software is intrinsically very social. Software likes to talk to other software. When software talks to other software it becomes more powerful, it becomes more interesting, it becomes more robust.

Not only does software like to talk to other software, increasingly it has to talk to other software. We have a composite of micro services, API stitching together those micro services and software talking to meet a need.

Thinking about all that very social software and thinking about it all talking to each other, the next logical progression is where does that software sit, especially business software or enterprise software. A few years ago the answer would have been pretty clean, pretty simple. All my important data, or my important software would be on my private WAN, my private data centres, my private MPLS network, the network and infrastructure that I as a large enterprise am paying tens of millions of dollars a year for. That's where everything that I cared about from a data or user or application perspective was.

Fast forward a few short years, and that's rapidly changing and it's about to hit an inflection point where there is actually more data outside of those heavily guarded WANs and private networks than inside and that's cloud and that's tens of billions of IoT devices, and that's mobility. That's where all the growth is kind of turbocharged or supercharged by AI, augmented reality, et cetera. If you think about that, it means the very design of the private WAN, everything we built, is negated. The private WAN is kind of built to create this fortress around this private data and applications. But now guess what? The more important, or the more interesting applications, the people are outside the fortress. They're on the net.

So when we say that the Internet is eating the world and the Internet is eating the private WAN I can be even more specific and say the concept of the private WAN is no longer needed. The design criteria are gone.

That said, and we talked quite a bit this morning, Mike and others mentioning in order to have change there needs to be some pain, there needs to be some time savings, there needs to be some business generated. There has to be motivation for change. So even if we believe the Internet is the next WAN and we believe that specifically the programmable Internet, which I'll talk to in a moment, is the next WAN, what's going to make us go there? Is there really the compelling reason, the pain, the benefit, et cetera? And I think there is.

Forgive me because I'm an engineer, I like to put things in kind of quick stacks here and make them look like a platform. But I think this helps me wrap my head around it. It's very simple, the whole Fourth Industrial Revolution, which kind of fundamentally changes everything in the world today is built on digital transformation. It's built on the fact that everything is going digital. Software everywhere. As I mentioned earlier, social software everywhere.

If software is everywhere and it's going to talk to each other, it needs distributed compute. That's the whole purpose of a network and sometimes I think we lose sight of why we're building networks in the first place. We put billions of dollars to make networks great and do this with networks and do that with networks. At the end of the day, the network is only there to make applications talk to each other across distance. That's it. That's as simple as a network is.

In our case though, in order to enable that Fourth Industrial Revolution and the digital transformation and the distributed compute the network has to work. It has to work. It has to be agile. It has to be secure, which we'll talk to in a moment. It has to be cost effective.

We, in kind of our rush towards digital transaction Fourth Industrial Revolution, we kind of take that for granted, or at least many folks take that for granted. Most of the folks in this room know that that's anything but the case. Today's network is not agile enough, cost effective enough, secure enough, performing enough, to be up to the task of the Fourth Industrial Revolution.

Specifically, if I had to sum it up in one word, I would say the Internet needs to be programmable. All the software it's supporting, all the applications it's supporting are agile, they're software, they change in the blink of an eye, or faster than that. With artificial intelligence, with AI, with cognitive compute we're actually faster than real time. We're taking actions based on events that we think are going to happen, that are predicted to happen. So we're past even kind of real time and we're into a mode where AI et cetera is moving us even faster. Therefore, the network better be programmable in order to keep track, or keep pace.

That concept again sounds simple, but to backup just a second, today network and applications don't even talk. They don't. They don't speak the same language. Applications do what they do in their user space. They pass the data down to the network and the network magically takes care of the rest and that's awesome and that's great and we wouldn't be here today if that wasn't the case. That made application development, modern application development, be as agile and cost effective as it has been because the developers didn't need to worry about the networking part.

Going forward with the Fourth Industrial Revolution and the fact that these networks now carry the world's most critical data and they're all interconnected and stitched together with micro services and APIs, we're essentially only as good as your weakest link, or maybe your least secure link. Now all of a sudden network needs to be programmable.

Specifically, Internet is not going to be reinvented overnight. We know that. There is unbelievable work being done in a lot of the IATF forums, a lot of other forums to improve protocols, change protocols, fundamentally change the way routers switches layer 2, layer 3 type devices work and that will happen over time, but it won't happen overnight.

In between now and then I believe what we need is this concept of a service connection platform and this is not just one monolithic platform. This is conceptual. This is what leads to programmable Internet.

So, the Internet itself, or any network, the job of that network is to connect nodes at the end of the day. Connect node 1 to node 2 to node 3 to node 4. Fantastic.

Connecting nodes doesn't necessarily connect applications or services. It doesn't necessarily give the application or service the compliance or the security or the reliability it needs. The only way that happens is if the application can actually tell the network what it needs and the network can talk back to the application and they can negotiate and manage and collaborate to actually get the application or the service to do what it needs to do. There is kind of four primary factors here.

One has been integration first. That's the communication between the network and the applications. That's APIs, that's SDKs, that's network as a micro service, that's the application creating the network.

As a quick example, just look at cloud compute. Just look at what WAS, EC2, Azure, et cetera have done. At the end of the day what they did is they abstracted compute and storage and they made it very easy for any of us to be able to spin up compute and storage on demand and spin it back down. It couldn't be any more agile and responsive and reflective of the actual application needs. We literally spin it up and spin it down at will and Amazon and Microsoft and everyone else has done a fantastic job in enabling us to do so. What if we could do that for networks? What if I could spin up or spin down a global network in the same way that I can do compute on AWS or Azure?

What if I don't pre-build the network but I have the ability to, as software, build those nodes on demand? What if I have a specific regulatory need or compliance need that means I need extra nodes in Germany in order to keep data in Germany? Or maybe I need a different pairing relationship in Southeast Asia to make sure that the data doesn't go across China. Or maybe I simply need more compute, more networking compute, in order to meet the performance requirements of the application that I need to deliver. That's what I mean by integration first.

The second part, security. Again, we talked a lot this morning about security and I agree with a lot of the sentiment expressed by the folks here.

What's interesting to me is software can take away a lot of that complexity. We mentioned the fact that today we do security after the fact and is very painful and it's reactive and we throw loads of people at it. As we said this morning, those people bugger it up. That is the reality in today's security and that's why it's painful and that's why quite frankly it doesn't get done.

But what if security was built in as software? What if we were secure by design? What if the complexity was taken care of by the software so that we weren't trying to take care of it after the fact? That's what these layers need to provide on programmable Internet. Need to be able to reverse the very permissive model of the Internet when that's necessary and it needs to do so programmatically on demand by the application.

The third point there on reliability, the Internet itself is not reliable. It's constructed of many unreliable nodes. A lot of unreliable software for that matter and as such, in the enterprise world, especially in the business world, sometimes we have this conception that well I can't put my applications on the Internet that need the five 9s of reliability, or need a certain level of performance because it's the Internet, it's best effort.

Somewhat true. However, again, let's just take a lesson from the world of compute. Look at what Google has done with their shard architecture. I do a Google search. There could be 20, 30, 40 nodes or servers or database structures down and it doesn't matter. In 0.002 I'm going to get the response back because of that distributed shard type architecture. We can do the same thing on the network side.

With enough resiliency we can get the reliability we need. We can get the reliability necessary for digital transformation and for Fourth Industrial Revolution. But it does

take that amount of resiliency and it takes that on-demand resiliency. The concept I mentioned earlier of being able to spin up network when it's needed and spin it back down when it's not needed.

Finally, one missing piece here is on-time delivery. Again, when we think about performance on the Internet, or really any network, we usually think about it in terms of bandwidth and pipes. Those are somewhat artificial constructs that really don't relate to on-time delivery. Again, the application is not telling the network I need these packets delivered in this amount of time. The application is handing the packets to the network and regardless of the size of the pipes they may or may not meet the time requirements of the application and whether that is due to any number of factors, because the network doesn't even know what the time requirements are to begin with and has no way to enforce it, it's best effort.

So there does need to be some constructs here by which the applications and the software can actually tell the Internet what their time requirements are. If I'm an FTP file transfer, I'm a data backup. If I can afford for that to go over a four-hour time period, that should be handled one way programmatically. If I'm a real-time full duplex video conference, then I have very different time requirements. And again, those things don't just depend on bandwidth, they depend on the timely delivery of the packets. So that's the fourth construct of what we'll see in these service connection platforms.

Finally, this is not revolution in the sense of we're trying to go overnight from a point way over here to a point way over there. We've actually seen some of this already start. We saw it first in the datacentre with software-defined networking. Software-defined networking defined in the original sense of software-defined networking.

The unbelievable work that folks like Facebook and Google have done in their data centres with software-defined networking where they abstracted out the control plane from the data plane to be able to get efficiencies and effectiveness and essentially make the private data centre programmable, which is what it comes down to. And look up some of the work they've done because it's awesome work and they've open sourced a lot of it. But they've done it inside the private data centre.

The SD-WAN folks did similar. They looked at what happened in the private datacentre and they said well, we can do that on the WAN. We can abstract out, we can take the control plane up a level and be able to control the data on the WAN. Again, did an excellent job and many from S&B enterprises are kind of benefiting from that SD-WAN concept. They're benefiting from it inside the private WAN, but they're benefiting from it.

The next step here is to take those same concepts from the private data centre, from inside the WAN and put them over the global Internet and make the Internet programmable.

At that point in time, we have a networking solution that can actually deliver the security, the performance, the reliability, the agility at the right cost basis of the Fourth Industrial Revolution.

**Manek Dubash**

Great. Thank you, Galeal and let's invite Alan Zeichick just to come down and ask a few pointed questions I think. Alan.

**Alan Zeichick**

Well not too pointed. Then we'll have some questions from you people as well. Thank you. So the first thing that resonates to me with what you were saying is we say the purpose of our IT infrastructure is to connect users to applications. We look at our network and it's designed to connect devices to devices. It sounds like what you're describing as the programmable Internet is designed to connect users to applications or to services, or services to services. What type of business opportunity does that create? By this I mean think about how every time there is a new type of whatever it is on the Internet, new businesses emerge to take advantage of it, to offer new services out to businesses or to other infrastructure players. Does the programmable Internet create new opportunities, new product categories, new service categories?

**Galeal Zino**

A great question Alan. So actually Alan's question is perhaps what motivates me the most. I can't wait to see what type of innovation developers are going to be able to produce when they're unshackled from the network. The network is handcuffs right now. It's handcuffs to software developers, IT administrators, enterprises for a security perspective, cost perspective, lack of agility. It's a handcuff. It's a separate handcuff.

You remove those handcuffs and you'll see innovation and I don't know where that innovation will be specifically, but I know it will come. The analogy I usually use is Twilio.

Twilio did similar. Twilio abstracted voice and SMS from the voice and SMS networking. They made it so that any of us could easy embed text message into voice in our applications without having to become voice engineers or SMS engineers and as a result, we saw innovation that we hadn't even thought of before.

Two-factor authentication is an easy example. Before Twilio came along two factor authentication didn't exist, using text messages for two-factor authentication. If you had to do that with a telco, if Uber and Airbnb and the folks that use the most two-factor authentication in the world had to do that with a telco, never would have happened. Twilio took the handcuffs of text messaging and voice away, innovation resulted. If we can do our job and take the handcuffs of the network away, innovation will result, business value will be created.

**Alan Zeichick**

Thank you. Another question then I will open it up to everybody here is you talked very briefly about SD-WAN and SD-WAN of course in many cases lets companies, organisations sort of choose between the public Internet and private MPLS or other circuits depending on applications. Is that a sustainable model going forward? As you talk about using kind of add additional reliability to the public Internet, what's going to

happen to SD-WAN? What's going to happen to MPLS and all these extremely expensive but extremely reliable technologies?

**Galeal Zino**

Great question. So the reality of it is, especially a large enterprise has a variety of networking needs. That's why we see hybrid networks taking place. It's not that I as an enterprise think that hybrid networking is cool and I'm going to go do it. I have business needs which dictate that I need to do some variety of hybrid networking. I might need MPLS there. I might need SD-WAN there. I might need Internet here. That need won't go away any time soon and these things take a very long time, especially in the large enterprise space.

At the end of the day, I think each of those enterprises is going to do what's best for their business needs. For sure, an increasing percentage of those estates will be Internet first. As we talked about, that's where the growth is. Whether we're talking 20 billion IoT devices, whether we're talking mobility, whether we're talking cloud, the massive distribution of compute in tomorrow's world it only lends itself to one network. Only one network in the world has the cost effectiveness, the scalability, the extensibility and the agility to be able to cope with that massive distribution. It's the Internet and how the enterprise works that in from a hybrid networking perspective will be different per enterprise.

The MPLS part of your question is easy to answer. That goes away. That's easy. The rest of it will look…

**Alan Zeichick**

Bad news telcos.

**Galeal Zino**

Yeah and I'm coming from a telco background. If any of us in this room were designing the network right now for our enterprise needs from the next 10 years, MPLS won't even be in the discussion. That part is easy. It's not software-defined, it's not agile and it doesn't have a cost basis.

## *Audience Q&A*

**Alan Zeichick**

Thank you. We have time for a couple of brief questions. I always say that, brief questions and brief answers. Put up your hand. I don't want to get in the way here.

**Joel Stradling, GlobalData**

Thank you for the presentation, Galeal. Joel Stradling from GlobalData. Could you describe your vision or the ideal within the programmable Internet and programmable WANs how the application-drive network and distributed computing will provide security because that's been a recurring theme today? Then also, how does perimeter security fit within that discussion?

**Alan Zeichick**

Or is there a perimeter.

**Galeal Zino**

That's right. So I think security may be the most interesting part of that whole conversation. There are two fundamental pieces that need to be put in place for that security. One is authenticate before connect. What I mean by that is today when I connect to the Internet, I'm connected. I'm connected and I'm able to do all kinds of things, maybe awful things, immediately. Before the network knows who I am or has authenticated me or authorised me I am connected. In tomorrow's world, it needs to be authenticate before connect. I need to be authenticated and authorised before I'm allowed to see those network nodes. It has to be that proactive.

Second, once I am authenticated and authorised and the network is convinced that I am who I say I am and I have access, then the question is well what do I have access to. Again, in today's Internet world, it's everything and that's been great for what's developed on the web et cetera. In tomorrow's world, the answer needs to be only what you need to access. It needs to be at least permission model. So we need the policy and the identification structures so that when I do hit that network I am authenticated, I'm authorised, and I'm only allowed to use very specific parts of that network and very specific might be by geography, it might be by time, it might be by application, it might be by IP address, comm port, comm user agent. It can be very, very specific and granular.

With those two constructs we then have a proactive element added that drastically reduces what has to happen on the backend. With those, we don't have to dump the legions of IT people on there to try to fix things after the fact.

We have a starting point that's here instead of here. That said, that's a starting point. Security is layered. It takes next generation identity, authentication. It takes co-operation amongst multiple partners and it's logical I think that Internet needs to be programmable.

**Alan Zeichick**

Great. Time for one question.

**Mike Spanbauer, NSS Labs**

Thank you. Mike Spanbauer, NSS. So regarding applications that are performance intolerant, we've encountered this issue for years. One of the concerns is of course

latency in round trips. So is it the application's responsibility or the developer's or the networks compromised to be able to overcome this between state that we find ourselves in. Love to hear your thoughts.

**Galeal Zino**

Yeah, Mike I think it's collaboration as we talked about earlier. You mentioned this morning that security is reactive and it needs to be proactive. That's one reason why security has often come at the expense of performance. And to some of the other points this morning, that's why security doesn't always happen because it's giving me some pain and I don't like pain. I don't like. I want performance and I might be willing to take a risk in order to get that performance.

So I think Mike when the applications and the network can actually talk and they have constructs to be able to deliver performance for the applications that need performance and they don't have separate security infrastructure that gets in the way, now we have a starting point to be able to deliver that type of performance.

**Mike Spanbauer**

What about (inaudible)? Background application compatibility is part of the (inaudible) to the industry. I don't know how we get to that.

**Galeal Zino**

Agree. Agree. We'll see innovation there with programmable networking, just like we've seen in modern application development, as we've seen kind of legacy applications be ported or extended via adapters, via APIs, via refactoring. I think we will see the same thing for legacy applications on the new network.

**Alan Zeichick**

Great. Thank you. Thank you, Galeal.

[end]