



Why We Aren't Prepared for What's Waiting Over the Horizon

Author: Marty Sprinzen, VantIQ CEO

Wrap your head around the following scenario:

An ambulance is racing down the freeway, transporting a suspected heart attack victim in critical condition. Long before the patient arrives at the hospital, doctors are remotely examining his condition, pouring through up-to-the-second data from sensors and video feeds transmitted from the ambulance over a 5G wireless connection. Even before he gets wheeled into the emergency room, the attending physicians already have a detailed picture of the problem and are ready to go to work. Or, even more exciting, the patient's life was saved by being remotely treated while in the ambulance.

Amazing stuff, but it's just the coming attraction.

We're on the cusp of a transformative era where the deployment of real-time, event-driven technologies turns vast volumes of data into actionable information, leading to fundamental improvements in everything from commerce to personal security. For example, if a child gets lost in Central Park, her picture could be automatically transmitted to thousands of video cameras equipped with AI recognition to enable searchers to find her.

Or think about filling your grocery lists with the help of intelligent shopping carts that guide shoppers through a supermarket, optimizing their route and making suggestions based on what it knows about their buying habits. (In fact, these intelligent shopping carts are already being tested in Israel and the results show shoppers using them are spending on average 20% more per visit.)

That's a glimpse into our future – if we do it right. But let's not get too far ahead of ourselves. As cool as it sounds – and it is – this sort of advanced functionality is complicated and requires a reliable infrastructure that can guarantee the kind of high-speed bandwidth for the transfer of massive volumes of information – which still needs to be built.

Gaining an Edge

One advancement that will help speed this transition is the emergence of multi-access edge computing (MEC), which is essentially a distributed architecture where compute and cloud-like capabilities get pushed out to the edge of the network. This will be key because the new applications we're talking about put extra demands on network capabilities, particularly when it comes to low latency, high bandwidth, and resource consumption. The telcos are best-positioned to lead the way here and we should expect to hear more from them over the next few years.

Think of the possibilities that would arise as massive amounts of data get filtered and processed on the edge in actual real-time as these systems react instantaneously to help manage critical events. Think back to our connected ambulance example. The system is communicating with the biometric systems monitoring the patient in the ambulance. If something about their condition changes, the emergency personnel in the hospital are immediately forwarded an analysis of the voluminous and complex data collected at the edge.

A lot is going on in the background as this deployment must support the dynamic movement of data in a distributed edge-enabled network. While the ambulance moves down the road, data sent from that vehicle must get transferred reliably. Maintaining ongoing contact with the ambulance literally becomes a matter of life and death where the system must guarantee that the information will get picked up by the next MEC down the road.

The Developer's Dilemma

When it comes to designing applications that make all of that work seamlessly at the edge, we immediately encounter two problems. One is a developmental challenge, the second an operational one.

From a development perspective, you need to tie everything together to create real-time systems that essentially sense, analyze, and act on situations of interest. That's not easy. Operationally, how do you deploy and then keep this edifice up and running? And how do you make sure that the infrastructure can handle the load as there's a need for more capacity?

Also, an event-driven application in a distributed environment (such as when a connected vehicle changes from one network to the next) needs to be able to share data across multiple administrative domains. That raises new multi-tenancy issues. Do we share data across multiple administrative domains? How do we do that? Am I sharing inside the company or outside the company? My point being, that in a distributed environment, these developmental and operational issues get extremely complex in a hurry.

Think again about the example of our ambulance passing one MEC after another as it speeds to the hospital. The MEC is running the logic – essentially serving as a mini-cloud – where it's analyzing the streaming data on the patient, such as an EKG. When the vehicle moves from one MEC to the next, however, what we refer to as the “state information” from the ambulance needs to get transferred seamlessly when the ambulance connects to a new MEC, rather than re-calculated from scratch.

Make it Simple

These types of applications are highly complex, and they won't work with traditional development platforms where a simple application may take months or longer to create. This is the third time in my career that I'm building distributed systems. I did it at Ingres and Forte – and now again at Vantiq – and I can say from experience that it's incredibly difficult. Things are getting too complex for us to hard code anymore. Even elite coders are discovering that this kind of task is near impossible and would require a series of discrete products to get the job done; and even then, it might take months to finish the project.

Frankly, we need to make coding a lot simpler – to find ways to help developers convert high-level operational processes into a framework for real digital applications.

One way to overcome this hurdle is to embrace an abstracted low-code framework that allows developers to handle the complexity that they'll face in these kinds of scenarios. These low-code tools allow for a more agile framework that developers can use to create solutions much more quickly. While different forms of low-code have been around for a while, the approach is fast gaining speed thanks to the market's demand for digitalization, which soared throughout 2020 as businesses accelerated their plans due to the pandemic. It also fits well with the need for speed found in mature agile DevOps practices.

How this unfolds in practice remains unclear, but the debate over the best approach must surface sooner rather than later. We're fast approaching a future where we'll need applications that can take advantage of the latest advances in IoT, AI, machine learning, and edge computing. And we'll be there in a hurry.