

IBM Podcast

[MUSIC]

PIPER: Hello, it's Andy Piper here, and I'm the WebSphere Messaging Community lead for IBM, based in our Hursley lab in the U.K. where many of IBM's messaging and integration technologies are developed.

We've been making some exciting announcements in the last few weeks regarding some of our technology in conjunction with our partners at Eurotech. And so, with me today to discuss these announcements are two gentlemen who I will ask to introduce themselves as I mention their names. So first of all we have Angel Diaz.

DIAZ: Good morning, good afternoon, good evening, everyone. Great to talk to you all, and I look forward to our chat today, Andy.

PIPER: So Angel, could you just tell us a little bit about usual role at IBM?

DIAZ: Sure. I'm IBM Vice President for Software Standards across all of IBM, and I spend my days at the cross section of technology, customer implementations and trying to get the industry to work together to make life

better for all of us.

PIPER: Okay. Thank you. And our other guest today is Arlen Nipper.

NIPPER: Hello, Andy, and thanks for having me on the broadcast. Again my name is Arlen Nipper, and I'm the President and CTO with Eurotech. I've been in embedded and end to end for about 33 years, and I was fortunate enough to be able to participate with IBM in some of the original developments around the MQ TT transport protocol.

PIPER: Okay. That's really interesting. So actually that's what we're going to be talking about today, MQ TT, which stands for MQ Telemetry Transport. So let's assume that I've never heard of MQ TT before.

Maybe we could have a little introduction to what this thing is, this protocol is, how it works, what it provides. So maybe, Angel, would you like to talk a little bit about why this is an important and useful protocol and where it's come from?

DIAZ: Sure, well, let's step back a moment. When we talk about machine to machine, we talk about technologies that allow both wire, wireless systems to talk to each other, right? [End user] devices usually, to capture sensor

or meter information, [a sense percent] such as temperature, inventory levels, things like that. And it all goes through a network. Right?

Now, let's look at what's happened over time. Over time, this notion of just single device, single endpoints, right, it's served us very well. But we've become more connected.

As a planet, we've become more connected. There's been an Internet of things that have started to occur. And the network of devices that are now able to communicate with each other and work with each other has really exploded.

So what has happened, Andy, and everyone out there, is that there's been a real need for a lightweight kind of publish/subscribe protocol to have more predictable bidirectional delivery of the messages. Okay?

So that we can start to move from simple point-to-point end-to-end interaction to more sophisticated network-based interaction across the network. So MQ TT, that is a protocol that actually allows exactly that: the publish and subscribe of messages and a predictable, bidirectional delivery of those messages.

Now, what's really interesting about this is that not only does it help with the actual connection of these devices,

but as you start to look at the type of analytics, the type of rules, the type of processes, the things that you can do now with that information that's being shipped around the Internet...

You can start to just imagine applications that actually focus more on the business processes themselves, not just the connectivities. And as a result, our networks will get more smarter, the way we work will get smarter and the planet will get smarter. So this is actually a really exciting technology and it's a real exciting time to be working on this stuff.

PIPER: Okay. So Angel's talked about the fact that this is a network protocol very good for small devices. Now, Arlen, you've been dealing, as you said, with embedded systems and small devices in different industries for a long time. Angel's mentioned this as a publish and subscribe protocol. Is there anything more you can add to tell us a little bit more about why this is a useful way of communicating between devices?

NIPPER: Sure, Andy. Well, actually, we started on the design of MQ TT 12 years ago. So what I want to do, is I'll tell everybody a little bit of the background of the evolution of MQ TT, where it came from and what its roots were, and then I'll follow up and expound on what Angel was

saying, because my view of the world was very, let's call it, embedded centric.

And when I came to IBM and started working on this and saw how IT addresses devices versus the way the embedded community I think ancestrally looked at devices and solutions is quite a unique paradigm shift.

So, first of all, I had been working in the communication gateway industry for many, many years mostly in oil and gas, and that's kind of where some of the naming conventions came from, because my background was in telemetry and SCADA systems.

And then in the mid nineties, early nineties, as you well can imagine, we did not have the networking connectivity that we have today. And so one of the projects I was working on was with Phillips Conoco on the new pipeline SCADA system where we needed real-time delivery of data and just by serendipitous coincidence got hooked up with IBM.

And what we tried to do was take at the time what was IBM's middleware, MQ Integrator, and at the time what I was doing in the communications industry talking over 1,200-baud dial-up lines and 300-baud dial-up lines and very bandwidth-constrained VSAT and tie those two together.

So when I talk about MQ TT, our five goals when we implemented it was, it needed to be simple to implement. It needed to have a quality of service data delivery. It needed to be lightweight and bandwidth efficient, because at this time bandwidth costs money. It had to be data agnostic, and it had to have continuous session awareness.

And what I mean by that is that it's all very well and good to have a device on the network and be talking TCP/IP and be event driven. But when you get to, let's call it, a mission-critical end to end, you need to know where these devices are and what their current state is.

So I think we did that, and I think that's why MQ TT has been around for as long as it has, is that we met those goals, and that's now what you see as MQ TT in the current specification.

Now, that's the underlying technology, but the more interesting aspect, for me, very pointed, Angel said, is that end to end up until now has typically been machine to machine, but that's a problem.

When we invent an embedded end-to-end device, the notion is well, you know, I invented the device, therefore I need to invent the protocol, therefore, I need to run on the IT side and invent an application that talks to my device. And what

you get is a point-to-point solution. It becomes point in time and it becomes frozen.

So what I learned in working with IBM on this project is by going with a publish and subscribe -- more of a three-tier middleware architecture -- what we're able to do is separate the implementation of the data producers, and they could be in the business side, it could be on the device side, and the data consumers.

So now I can have customers go out and write very interesting applications on an end-to-end device, publish data and be completely oblivious of who is going to consume the data. And then on the other side, you know, I can have people consuming that.

PIPER: So we're really talking about a nicely decoupled system with potentially a lot of small devices just emitting small pieces of data, and we don't mind what format the data's in, it's all about having a protocol that's reliable and appropriate for this kind of, these kind of networks.

Okay. So machine to machine, end-to-end messaging. It seems to have been around for quite a while. We'll talk a little bit about some of the places where the MQ TT technology's particularly being used.

But the announcements that have been made in the last couple of weeks, Angel, I think are really looking to expand and drive this kind of ecosystem. Can you talk a little bit about what's been announced from IBM and Eurotech in the last couple of weeks?

DIAZ: Sure. So the Eclipse group started what they call an industry group around machine to machine. And most recently we've got a focus in that group. IBM has donated its MQ TT client source code to that community group. It actually comes from a connection pack that we had shipped with our MQ products, but now what we've made available for all of our partners and clients everyone to use to start building.

And really the objective is to grow the people who start to take advantage of common protocols, common standards, open source technology, to accelerate the rate and pace at which people get more connected and drive value.

We want intelligence to be infused into systems and processes. These are the things that make the world work. By getting different industries -- not just technology vendors but getting people who actually use the technologies -- we'll be able to put this type of intelligence into things that you would never recognize as computers: cars,

appliances, roadways, power grids, clothing, even natural systems such as agriculture and waterways.

We've been talking about this for a while. I think in 2008, we've been talking about what it takes to build a Smarter Planet. This type of technology, this type of ecosystem that we're going to start to build within Eclipse is going to help deliver that and bring the industry together. Very exciting.

PIPER: So, we've certainly seen a lot of success for the Eclipse Foundation over the course of the last 10 years that that's actually been running now, so it's really exciting to see this kind of code ending up there. Not just a standard, which is out there or being standardized through an open process but also to actually have some source code out there as well.

And I know the Eurotech are working very closely with this as another company called Sierra Wireless, who are really participating and collaborating around this end-to-end space.

We've talked about protocols, we've talked about a load of technology and source code and other things. Let's talk about some real use cases. We've kind of thought about some of the possibilities, but certainly, Arlen, you've been

building these kind of solutions with different companies for a long time. Is there anything you can...any examples you can talk us through very briefly?

NIPPER: Sure, Andy. Well, as I mentioned kind of in the introduction, kind of the genesis of MQ TT was actually came out of the pipeline industry. So one of the use cases I'll go through here very quickly is how we use MQ TT to control the Phillips Conoco pipeline system.

And so, in this type of a control system, again, let's go back to my data producer, data consumer model. A lot of the equipment, the sensors, whether it be equipment status, pump status, bearing temperatures, flow rates, valve statuses, tank levels, you can imagine all of the sensor data flowing on a pipeline system.

And at the time there was one very important data consumer, and that's called the SCADA host system. And indeed he is a very important data consumer. But by using MQ TT and brokering technology, we were able to publish real-time sensor data and basically that was the data producers. And the data consumer -- let's call it the big data consumer -- was the SCADA system. So we were able to deliver real-time events to that consumer.

But the more interesting thing, and the reason we

implemented this, was you can imagine all of the other business intelligence that could be gathered from that information whether it be the maintenance department, whether it be the accounting system. You have real-time product information -- you know, I just sold a million barrels of crude oil and I want to be able to act on that.

So we got the system up and running, and then all of the other business data consumers could tap into that using this publish and subscribe protocol, and they were able to take that and really drive value.

So that's one good example. We've used this MQ TT in gaming systems, we've used it in medical systems. Really it applies across a large portion of the types of end-to-end applications that Eurotech gets involved in.

PIPER: That's really cool. I mean, I'm very familiar with a number of solutions that IBM's worked on as well. And I know as well as the medical example that you've kind of hinted at there, one of the really nice examples is really a true smart grid solution, a true smart metering solution...

...where a particular provider in the United States is able to provide their customers with a solution where their homes can be instrumented, with their permission, and using the MQ

TT very lightweight, very low-bandwidth protocol over a Verizon 4G network then they can actually send commands to the consumers' homes to have devices switched on and off, for example.

So it can be very, very power efficient and actually beneficial both to the power company who want to control their grid usage more effectively and also to the consumer to reduce their rising energy costs. So thanks for that example, or those examples, Arlen. They're really helpful.

Hopefully that's been an interesting introduction to the topic and the exciting end-to-end space. I think it's a very different and new space than many people, particularly developers, who may have been focused on, for example, the enterprise space. But these systems, I think, are increasingly becoming part of the Smarter Planet story, the need to bring in data and analyze it and respond to different events.

So there's certainly a number of places to come and interact. There's the mqtt.org Web site, which is the community site for the protocol. Really it's a vendor-neutral site which just looks to provide links to different implementations of the protocol, because it is under an open specification. And also, discussion groups and ways to interact.

IBM has, of course, Web sites providing information. Eurotech has some information as well, and there's some YouTube videos that will add to the show notes for this particular podcast. So, Angel, I'd like to thank you very much for your contribution today.

DIAZ: Thank you very much, Andy.

PIPER: And Arlen, really appreciate your time, especially as someone who both helped to invent the protocol and who has worked in this space for such a long time.

NIPPER: Well, thank you, Andy. After 12 years of seeing it implemented, I'm really excited about these new opportunities. I'm excited about the new M-to-M work group in Eclipse. And I think this is going to be really cool.

PIPER: I agree with you. Okay. Thank you very much. And that's the end of this particular podcast.

IBM Podcast

[MUSIC] [END OF SEGMENT]