

01:47

Speaker 1
You.

01:53

Speaker 2
Episode 18 of the Human and Machine podcast. My name is Jaco. As always, I'm your host with Lenny, my co host. And we're back from a fantastic trip to Cape Town over last week. A week before last week. We missed last week.

02:07

Speaker 3
Yeah, definitely. Cape Town was very kind with the weather, we.

02:12

Speaker 2
Too much good food and even better wine.

02:13

Speaker 3
Yeah, we actually saw the mountain for the whole week. So, yeah, it was a good week in Cape Town.

02:17

Speaker 2
Yeah, just a fantastic trip. And I think Cape Town, the Western Cape region, as far as manufacturing and innovation goes, there's some really exciting things happening in the Western Cape. A lot of young engineers, young companies doing some incredible things with ignition, other platforms and IoT, I suppose so. Really, really impressed by the Western Cape and a lot of good work happening over there. In our last episode, we spoke with Brian Pinocch from Mimecast. Just really about cybersecurity and the evolving threat landscape and how manufacturers can increase cyber resilience. If you missed that episode, it's well worth a listen, as Brian shared some really insightful facts and also guidance on how to become cyber resilient. And I think the primary one for me there, Lenny, was the fact that email is still the number one attack vector.

03:03

Speaker 3
I think that podcast scared the living crap out of me.

03:06

Speaker 2
As it should, I suppose, as it.

03:07

Speaker 3
Should being cybersecurity and just how easy it is still for infiltrations to happen.

03:12

Speaker 2
Yeah, that's scary stuff.

03:13

Speaker 1
Cool.

03:13

Speaker 2
So this week we have, definitely backed by popular demand, we have Walker Reynolds from intellect integration and 40 solutions based out of Texas. We ran out of time with Walker during our first recording. That's obviously the one where we got into oee why it's important, why it's still relevant. Great discussion. We did run out of time, and we desperately wanted to get a view of what Walker. It was a little bit of a carrot that he threw in right towards end of the conversation where you mentioned the unified namespace. So we really

wanted to get an understanding of what the unified namespace is and also what Walker describes as the holy grail. And Lenny, I know that's something that's dear to your heart as well.

03:52

Speaker 3

No, definitely. I think for me it's twofolds. If you look at the kind of standard automation pyramid and how integration we used to do, integration between the layers, what now for 2015 years? If you take one block out of the integration pyramid, then you're pretty much stuck. You have to pretty much redo everything again. And I think with the concept of the unified namespace, it definitely shatters. That bounds. Another point for me that's very dear to my heart with the unified namespace is especially with us at element eight. I mean, we distribute three products. Ignition canary flow. Yes, there's overlap, yes. Some of us. Dashboarding capability. Yes. Other has as well. And the unified namespace is really a concept where you can build a dashboard tomorrow in ignition. If it doesn't work for you try build it in the other solution.

04:42

Speaker 3

The point is that you do not throw your work away. That KPI and stuff that you've done is available in the namespace for any other piece of software to actually consume it. And I think that's great.

04:52

Speaker 2

Okay, well, there's two things for Walker to address already, so we have walk online. Walker, really good to have you back on the podcast. Thanks for taking the time again to join us. Early morning your time again.

05:01

Speaker 1

That's right. Thanks for having me again, guys. And this time it's 730 local.

05:06

Speaker 2

All right, so it's a late morning for you.

05:09

Speaker 1

Yeah, that's right.

05:10

Speaker 2

How have you been?

05:12

Speaker 1

We've been blessed. We've been incredibly busy. A lot of new engagements with really large enterprise clients, and we moved into a new office. In fact, we just completed that move. This past weekend is my first day sitting at the desk in the new office. So things have been incredibly busy, but also at the same time trying because of obviously the state of the economy and there's a lot of instability in the market. But with instability comes a lot of opportunity. And so I think everyone in our industry has reason to be encouraged, not discouraged.

05:59

Speaker 2

Absolutely. And I think busy is good. I mean, just being mindful and grateful for being busy where so many folks are not busy at the moment. So that's something to be grateful for. So Lenny already opened up with two very interesting departure points and notions there. I don't know if we want to kick off, right with those around the notion of an IIoT platform or the promise that an IoT platform delivers, that there is a unified namespace.

06:29

Speaker 3

For me, I want to chat a.

06:30

Speaker 2

Little bit about scaling as an important consideration, but yeah, really, the unified namespace is something that I was not really aware of, at least not per your definition. So we'd love to maybe kick off with that.

06:44

Speaker 1

Sure. So before I start, let me start with, I want to lay a basic premise, and that basic premise is all organizations are trying to achieve the holy Grail, right? So if you think about what the holy Grail is, if you were to take most chief executives of manufacturers, whether those are Fortune 500, global, one hundred s or mom and pops. The holy Grail is they want a fully integrated business, that is a fully integrated business made up of digital factories. And why do they want those factories to be digital? Because it's much more efficient to be digital. A. You have a higher rate of accuracy when record keeping, when event tracking is all digital, right?

07:31

Speaker 1

You get the event from the horse's mouth, and then you store it digitally, and you can be certain that if it was integrated correctly, that the data is accurate. So, number one, it's accurate. Number two, it's much more efficient. Humans are slow at data collection, right? Humans are really great at innovation, but we're really bad at very repeatable, at all of the repeatable tasks. And by bad, I mean we're not efficient at them, right? We get bored, and with boredom comes inefficiency. So the basic premise is a fully integrated business made up of digital factories. And what that means is that everything and everyone is plugged into a network. All the layers of your business are integrated, and they operate based on data and information from all of the other layers in real time. This goes to the automation pyramid that Lenny mentioned.

08:24

Speaker 1

The stakeholders all know the state of the business in real time. The state that they care about is, what is our cost? What's our profit? Where are we on schedule? Where are we to our plan? Where are we to our forecast? And there's one stakeholder that's correct, right? Stakeholders know the future state of the business in real time. This is the game changer piece, right? Forecasting. Anyone in business knows that forecasting is essentially the most difficult component of your business. Forecasting future state is very difficult. This is one of the reasons I love flow as much as I do. And this is the reason I mentioned it in the previous podcast, right, was that flow is very good at predicting future state based on current state and past performance, right? I mean, that it's baked into the platform, which is. That makes it different.

09:17

Speaker 1

The next component of the Holy Grail is organizations. These digital factories are leveraging technology, specifically machine learning and artificial intelligence, to collect and analyze data and information. The machine learning predicts future outcomes based on past patterns and current state. This is something that flow does, and artificial intelligence consumes many of the determinations that machine learning predicts to recommend operational adjustments to improve future outcomes. And then stakeholders, the human beings, execute or not execute recommended operational adjustments. That's the holy Grail. If that's our premise. And it doesn't matter who you talk to, if you look at predictive analytics, if you look at just the maintenance component of running in manufacturing, predictive analytics has been a thing, has been a dream for 20 years. Right? I want to predict failures, right? I mean, my entire career, people have been trying to do that.

10:18

Speaker 1

But how many organizations really do predict failures accurately? Not many. Right? They start with a great idea, but they build it on top of a flawed infrastructure. Right? Because again, the industry 30 way of integrating a business is plugging one layer of the stack into the next layer of the stack and plugging that layer into the next layer. The analogy we use here and Intellik is that's the equivalent of building a wall on top of a couch. You buy a sofa, you put it in your living room, you decide you want a new wall, and instead of moving the sofa out of the way, you build a wall on top of it. Well, the problem is that sofa becomes integral to the architecture of the business. And so if you remove the sofa, everything comes crashing down, and you got to rebuild the wall.

11:13

Speaker 1

That's exactly what Lenny was talking about, right? When you make a piece, a solution, a fundamental

component of your business that other things within the business depend upon, then you can no longer remove that solution, even if that solution doesn't do what you need it to. And that's the point. So the unified namespace is the idea that what we do is we create a single source of truth for the entire business. Now, someone might ask, well, isn't that the digital twin? And the answer is, no, it is not the digital twin. It's similar. There's a similarity between the digital twin and the single source of truth, but they're not the same thing. And I don't want to go into the technical or theoretical elements, what makes a digital twin different from a unified namespace? But the idea is that we do two things.

12:10

Speaker 1

Number one, when we get everything and everyone plugged into a network, the next thing that we do is we treat everything and everyone as a node in the network, and then we create a single source of truth that everything and everyone publishes into and consumes from in order to take data and produce information. That is the unified namespace.

12:35

Speaker 2

Yeah, and you mentioned the truth. I think we may have mentioned in the previous recording that we've had a couple of instances when the truth is exposed. We've often heard, whoever we've been engaging with at that point, we've heard that, oh, we actually don't want to see.

12:53

Speaker 1

The truth, or you're wrong. That can't be the truth.

12:58

Speaker 2

Exactly. Let's rather give us the ability, if it is the truth, then let's rather give us the ability to manipulate it somehow before it's seen as the truth.

13:06

Speaker 4

Correct.

13:07

Speaker 2

And I think there is also a level of organizational culture that's got to be driven around that. At the same time, outside of just the tech and what the tech and the data shows us, I think there's definitely some alignment from all the different stakeholders involved to get to that one view.

13:23

Speaker 1

Well, and Lenny brought this up in the last podcast, right. And that was, organizations need to abandon the idea that data and information needs to be treated on a need to know basis.

13:37

Speaker 2

Right.

13:39

Speaker 1

Part of the philosophical change, if you're a leader, an industry leader, you're either an executive, you are a director, or you're a plant manager, and you're listening to this podcast, and you're one of those people who generally reads the Gartner Magic Quadrant every year. I'm going to give you two pieces of advice. Number one, stop reading the Gartner Magic Quadrant, because all of that was just paid for. Everyone who shows up on there paid, quote unquote, advertising to get their names in those quadrants. Number one, and anyone who doesn't pay advertising doesn't get their name in the quadrant. So it's essentially useless. Number two, you have to change your philosophy. And the first thing that has to change is you have to treat data and information as knowledge, and you want everyone in your organization to be as knowledgeable as humanly possible.

14:35

Speaker 1

And the first thing that you have to do is get them the data and information they need when they need it in the form they need it, where they need it. For the nontechnical leader that I'm speaking to right now, what you need to do is create the smartphone for a smart factory. That's what you need to do, and plug all of your people into it. If what you want to do is be the executive, if you want your legacy to be that, I was the executive. I was the Lee Ayakoka of the late 70s who turned Chrysler around. If you want to be Elon Musk, who created Tesla when everyone said it was impossible, or SpaceX when everyone said it was impossible, or if you want to be bezos, who said, how are you going to create a trillion dollar company selling books?

15:19

Speaker 1

Or you want to be Steve Jobs who says, who the hell is going to buy a smartphone? Nobody's saying they want a smartphone. Or you want to be Henry Ford, who created cars when everyone wanted a faster horse and buggy if you want to be those people, then the first thing you have to do is change your philosophy.

15:39

Speaker 2

Yeah, for sure. And I think what you mentioned also speaks to the scaling that I mentioned, that, I mean, it makes 100% sense. Like with most things, digital scaling is a consideration for growth. And when organizations implement these new IoT solutions, they want to add new applications to the system, need to build new connections between the software and all these other interconnected systems. What does all of that mean for integration costs and deployment costs? And that very much speaks to what you just mentioned. And it's something that really has to be considered from the word go and not be considered as an.

16:16

Speaker 1

Bring. This is where I want to bring Michael in at the end of my comments here. So one of our lead engineers, so one of our senior engineers, and he is the other half of our architecture team at Intellik integration. So the two solutions architects that we have are myself and we have a guy named michael Walker who's on the call. Michael. And I want michael to chime in. Know the concept of the unified namespace came out of my head, right? I mean, I know other people probably thought of the same thing, but I wasn't taught the unified namespace. The unified namespace was something that it evolved in my head. It was one of those things like, why do we do it this way again?

16:59

Speaker 1

Why is it we're not just putting everything, why do we keep assuming what it is people want to know when we're integrating? Why isn't we're just not giving them all the data in a place to use later on if they want to? And why aren't we putting everything in a unified namespace? Why aren't we putting it in a central location? And people have tried to do that, right? And the idea was, hey, let's use Osipie, or let's drop it into a database and this goes to the scaling thing.

17:25

Speaker 2

Well, that's another good question. That's another good question right now. But isn't the historian, is that a unified namespace?

17:34

Speaker 1

Definitely not.

17:35

Speaker 3

Even worse. I mean, we see it every day. Sorry, Walker, we see it every day where people say, I'll just let this Iot little sensor push data into my azure data mart, or wherever it is sits in the cloud. And they think that is now a platform where I can now take data out for machine learning, et cetera, which is definitely not the case.

17:56

Speaker 1

And what I would say is what I would do is recommend for anyone who believes that's the case. What I recommend is that they go to high bytes YouTube channel and watch the webinar that they did earlier last week. So it would have been November of 2020. Go watch the high byte, because we talk about why you can't do that. It comes up in there. When it comes to machine learning and artificial intelligence, normalization of data is a critical component, right? And that's the reason you can't take events with all various timestamps out on the edge and dump them into a data lake and expect to be able to run linear regressions against them. It doesn't work. It's the reason machine learning fails. The first thing you have to do with events is you have to normalize them.

18:43

Speaker 1

That is, you guys already know this because of flow, right? You already know. Part of the reason you guys do the buckets of time is so that you get the data normalized. I think you guys call them, or you call them measure, you call them buckets, right? Yeah, the reason you do the. Right, but I want to talk about scalability, right? So the reason you can't dump it in Osipie and the reason you can't put it in a database is it won't scale. There are too many data points, there are too many transitions. It won't work. That's number one. Number two, there's engineering that goes into that, what you have to do in order to be able to create a unified namespace. So let's define what the unified namespace is.

19:27

Speaker 1

Unified namespace, number one, is a single source of truth for all data and information in your organization. Think of this as a file share. So when I go into a file share, the corporate file share in my company, it is set up with a structure so that I know how to navigate to the folder I need to get to and retrieve the document that I want to get. And anyone else can follow that structure and do the same thing. When you add a new employee, that employee adds their folder to the correct place in the namespace and they put their information and data in it, right? In a nutshell, that's what the unified namespace is, with the exception of the unified namespace, is really structure and events.

20:10

Speaker 1

So when you think of an MQTT broker, what we do is we create the, if we use MQTT as the technology, which it doesn't have to be, we just recommend that it be MQTT. The structure is ISA 95. Why? Because that's how every organization either does or should be structuring their information. ISA 95 is enterprise site, area, line, cell. Right. And then what we do is we extend that. We create namespaces for all of our applications. We create namespaces for all of our functionalities, and we put them at the level of the organization that they apply. So I may have an area of the namespace called OEE, and so I may have, under the enterprise layer, I may have an OEE namespace, and all of my site namespaces. And underneath site, I may have an OEE namespace and all of my area namespaces.

21:11

Speaker 1

As I go into that namespace, I'm able to navigate through and look at all of the contextual Data, all the stuff, all the information we created from data events that we see in our equipment and our people. And I'm able to create new stuff and put it in a place where other people can go grab it. That's what a unified namespace is. To create a unified namespace, you have to connect all the things that have unique namespaces. So a PLC has its own unique namespace. An application has its own unique namespace. The unified namespace tells that node how to organize its unique namespace and where to publish its data. And when you organize it and publish it, now it's available for everyone to use. We use protocols. We have some requirements, right, in order for this to scale.

22:02

Speaker 1

When we talk about scalability, you want it to be as lightweight as possible, and you want it to use as little memory on a device, and you want it to use as little bandwidth on the network as humanly possible. In order to do that, you want to use report by exception. So that is, we only want to send information that changes. We want to use edge driven. That is, we don't want to have to request for changes. When a change happens, we want it pushed into the unified namespace, and we want it lightweight and open architecture. The reason the open architecture piece matters, because if what you want to do is change out, let's stay continuous here. So were using ignition, we're using canary, we're using flow. Here's something that all three of them have in common.

22:53

Speaker 1

You can build dashboards in all three of them, okay? And all three of them are good at building dashboards. But where you build the dashboard is a function of the information you're displaying on that dashboard. There is a place where flow is the appropriate mechanism to create the dashboard. There is a need where canary is the appropriate place to build the dashboard. And then ignition would be the place where you build the generic dashboard. Right. Ignition is not great at anything. It's good at basically everything. Right. But it's not great at really anything. Ignition is a jack of all trades. It's the swiss army knife. Right. Flow is not a swiss army knife flow. It's a japanese chef knife. Right. That's what it is. And it has a very special, specific thing that it's really good at. Same thing that canary is. Right.

23:56

Speaker 1

The point is that I could initially build my dashboard in canary, be unhappy with it. You know what? That's data. That's generic, and it needs to be in our generic tool for building dashboard. And I could literally just kill the dashboard. I could take a screenshot of it, kill it, go over and just recreate it inside of ignition. And there's no additional integration needs to be done whatsoever, because all three of the platforms are consuming data and producing information into the exact same namespace. That's open architecture.

24:34

Speaker 3

I love that idea that if you are consuming data to produce something or show something on a different dashboard and you're doing it three times, then you're doing it wrong. I love the notion that if you've developed unique content for your specific application to push it back onto the unified namespace, there's other applications, there's other solutions that want to get that calculation or that piece of data. And I think that's a very important concept of this, is to have that sharing ability between who's consuming and who's producing data, and it should be available for everything. Right.

25:09

Speaker 1

And I want to take it one step further, and this is where I want to bring Michael in. Okay, so what we've talked about is the specific. We've talked about capability, right? How the important of architecture is with a specific capability. That is, I don't want to paint myself into a corner. If my capability is, I want to create an efficiency dashboard that predicts future efficiency on this shift. If I create a unified namespace and I create a technology stack that is built on MQTT and report by exception edge driven open architecture, I can plug many solutions in. I could plug in Canary Labs historian. I can plug in flow software. What do you guys call Flow? What is the flow solution called?

25:59

Speaker 3

Information platform.

26:00

Speaker 1

Yeah, information platform. And I can plug in ignition. I can plug in all three, and all three of them can produce and consume from the exact same namespace, and I can move the capability around between the three of them. But I want to take that one step forward. So now I'm talking to the systems integrator, and I'm talking to the plant manager, and this is where I want to bring Michael in. This also extends to your projects. Okay. So a couple of years ago, and it'll probably take me ten minutes to do the background here. And this is important, though, and I think it'll be valuable to everyone. A couple of years ago, literally, it was about 26 months ago, were looking for a plant in Dallas that we could use as our example facility.

26:48

Speaker 1

Instead of us going into conversations with clients and trying to sell this on a whiteboard, right? What we decided to do was, you know what? We need a plant to show people that's what we need. We asked the question, what type of plant do we need to do that? What do we want to get out of that? Well, what we want is a plant that basically lets us. We want them to have no digital transformation whatsoever. We want them to be purely based on paper. We want them to have disparate systems. So that is many different types of PLCs, many

different types of software, all that kind of stuff. Right? We want them to be in a regulated industry because we don't want regulation to be the objection to not digitally transforming. And we want them to let us do it our way.

27:36

Speaker 1

Basically, we want them to get out of the way, okay? So that we don't have any of the political inefficiency. And what happened was about 26 months ago, just through sheer luck, that plant called us and they needed some help. They had decided they wanted to sort of digitally transform. Their vision of digital transformation was, let's connect all of our equipment using an OPC server, and let's build a custom web application that makes a dashboard, and it gives us basically three KPIs. That was their vision. They wanted to use Kep server and they wanted to use a data logger. So they brought us in, myself and another guy named Matt Olsen, who's our engineering manager. And Matt and I went in and we started working on this project. It took us a couple of weeks.

28:23

Speaker 1

We got all their equipment connected and all the network address translation done. We installed Kep server, we installed the data logger, and were working with their engineer. And when we got all done, we did what they asked us to do. We said, hey, you know, what we want to do is they had a printing press. So what this company does is they make packaging for food and beverage. They make chip bags, basically. So the raw material they bring in the back door is film. And what they ship out the front door is bags or rolls that could be made into bags. So it's a very complicated process. They've got an art department. They got all layers of approvals, and it's a very low margin. They do very short production runs. And what I said to the engineer, know, hey, his name's Mike, actually.

29:11

Speaker 1

Hey, Mike, shout out, I'm going to send this to you. I said to Mike, I said, hey, Mike, would your ownership, would your board of directors be interested in letting us do a pilot on that printing press right there? And it was like a multimillion dollar printing press, state of the art Siemens motion control, the whole DH 485, super high end stuff. And he said, well, what would you do? And I said, well, what we would do is show you what's possible. That's what we would do. And I said, and I'll pay for it myself. You don't have to pay anything. We'll do the pilot. It'll probably take a couple of months, maybe two months. We'll give you 30% of what's really possible, but it'll give you a taste. If you like it, you buy it and we do the rest of your facility.

29:55

Speaker 1

If not, we rip it out and we leave. And he said, well, let's go talk to them. So we did. We went in this presentation, and I said, I'd like to do this. And the leadership group said, well, what's the catch? And I said, there's no catch. You guys employ more than 100 people here in my hometown. These are people who all have middle class lives, they have mortgages to pay, and I want to make sure that you guys stay here. And I said, but there's two caveats. Number one, we're going to do it on our timeline. And number two, we're going to do it our way. That's why I'm paying for it. And they said, okay, sounds good. And then the last piece was, if we do this, we want to be able to share this plant with the world.

30:39

Speaker 1

And they said, sounds good to us. And so, long story short, we took two months and we did some key elements, like we digitally transformed their job jacket, which is a highly complex. The way that they build the job order when it comes in. We digitally transformed that. We installed Oee using our MES 4.0 platform. One of the first things that they weren't calculating OEE, we integrated with a shop floor system where they were taking inputs from operators. And then what we did was data analysis. We found that there were some databases running on the press that they didn't even know about that contained historical production information from the past. And so eight weeks later, when I presented, we really focused on three things. What does digitizing. What's the difference between digitizing and digital transformation? So we highlighted that. We digitized your job jacket.

31:40

Speaker 1

We digitally transformed your press. Okay, number one. Number two, we gave them analysis of two things. Number one, who were their most efficient operators. And number two, the flaws in collecting downtime

information and waste information from human beings. Because we were capturing the actual events from the machine and were comparing them to the ones that were being reported by the operators. That was number two. And then number three, what we did was we tied recipe set points on that press to historic data in the past that they didn't even know they had, that was running in databases on the machine the OEM never told them about. And when we showed this to the leadership group, they nearly fell out of their chairs. And two years later, we have fully digitally transformed their entire facility. Michael took over the project in January.

32:36

Speaker 1

And what I really want to highlight here, what I really want to highlight is we use the unified namespace to easily extend new capabilities. So this all started with essentially MEs and I don't want to say recipe management, but recipe analysis. So MEs and recipe analysis on all of their and. But since then it has extended far beyond that. And with that, what I want to do is bring Michael Walker in, who's one of our solutions architects, and have Michael speak to. How did this really change? How does this change the approach of doing projects in a. And specifically, what's the role of the unified namespace in saving the cost of integration and speeding up the time to value for these new capabilities? Michael, please.

33:29

Speaker 2

Awesome.

33:30

Speaker 4

Got it. Yeah. So just to give a little bit of my background, I primarily focus most of my career around MES. So a lot of database front ends to record production related data, OEE, traceability, quality, and then more or less correlating that with what typically surfaces out of a control system. So when I joined Intellic, the idea of the unified namespace, especially around MQTT, I knew about the protocol, had used it, but to be honest, in my home and then also in a couple of production environments, but it was all around just getting access to real time process data. And so Walker's idea of leveraging MQTT as a means to integrate, a lot of times. What is this point to point integration? I want to download orders from my ERP.

34:30

Speaker 4

I want to download bill of material from my ERP, reach into another system to pull out a product specification data set that we're running right now on the production floor. So that's where the shift change happened for me is moving from leveraging MQTT to surface real time process data to leveraging it as more or less an infrastructure that we can integrate any type of data. So when I took over this particular project, Walker's been explaining in January, we had a lot of challenges. We were showing them data, and the more data that was being seen, the more they wanted to do with it, specifically around things like their standards. So the big thing that the light bulb hit for them, that came off for them was, we are massively wasting setup time. It's not accurate. Our standard seems to be way off.

35:36

Speaker 4

So this particular customer came to us and said, here are new standards for this product running on this line, and this standard will change based upon are we moving from one product family to the next? And so we were able to integrate our system with more database tables that they had exposed to us. And leveraging the unified namespace, basically reach into that system, bring over these data sets, and match that to all the, I guess, tags, if you will, in ignition, in real time, consume that data set and calculate what OEE should be based upon all of their updated standards. And so I say all that to say is leveraging MQTT as a protocol to bring in all these external systems, not just the real time data, was the shift change for me that made it real. The unified namespace can work.

36:46

Speaker 1

Sorry, can you speak to the art part? Because I wasn't part of architecting the art piece. All I know is that the art group was a complete data silo. There was really no one else in the organization was connected to it. Can you explain? Because I think this is the one that the listeners will connect to, is the idea that we're taking, I think, on the plant floor, most integrators are accustomed to being able to see the state of all the equipment. Right? That doesn't mean that all plants are like that. But I think we're never surprised when we see the state of equipment

kind of unified. What we're surprised is when all of the ancillary business processes that are generally stuck in data silos become fully integrated with the business.

37:42

Speaker 1

Can you speak a little bit about the process working with the art group and how we essentially just extended the unified namespace to include all of their functionality, which then exposed the state of art to the rest of the organization. Can you speak to that a little bit?

37:59

Speaker 4

I sure can. So the art department, or the process of making artwork that gets printed on flexible packaging finished. Good. It has a massive amount of complexities that are involved with it. And so this particular customer, when they would receive an order, their art department had built processes that they follow. They were using Monday.com as a project management board to track the statuses of all of their art jobs that they had in process. And then they were manually interfacing back to their order entry system to basically indicate, hey, this job is done. You can pick up the finished good of our department, which is basically plates that get mounted and go into the printing press. And so what were able to do is extend the unified namespace into that department. And so architecting basically a new database that integrated with their order system.

39:16

Speaker 4

And now instead of going to Monday Excel, various other art specific systems to generate the digital artwork and the physical plate, we are now basically tracking through a SQL server database all the statuses of their outstanding artwork jobs and then also going back to their finished good item related to the order that they received. What are all the parameters that are going into that? So how many colors are on it? What's the width of the plate that's being produced? So hundreds of various different parameters that are going into defining what that artwork is. And so to take it a step further, after we built onto the one database that's now tracking that, were able to, when an order hits the floor, bring that data into the unified namespace and then expose it in real time.

40:27

Speaker 4

So if an operator, for example, if the artwork is supposed to run on a certain width, roll at their printing press, now we can go and say, okay, here's everything that's at the staging area available to run on this press. But based upon the artwork, which is now integrated with unified namespace, we're limiting visibility into what roles you can use based upon the width that the art makes up. So those are previous specifications that were being manually managed before.

41:01

Speaker 1

And yako, this is an important point because we get this question a lot. Well, does that mean if we're going to use the unified namespace for events and structure, right, does that mean that we're going to eliminate point to point connections between pieces of software? 100% the answer is no, you're going to eliminate many of them. But we may still have point to point connections where appropriate. But what we won't have is a query that is run from one node against another node that we don't put the result set into the namespace. So while we have SQL backends, when we fire an event to run a select statement, that select statement is run through the unified namespace. That is the data set the result set lives in the unified namespace for another consumer to see, for the historian to store.

41:55

Speaker 1

So that one of the implications is that you can look at when a query was run. We can recreate what it is an operator was actually seeing on the screen because we did that, because we have structure and events run through a common namespace, a unified namespace that is structured, we can go back in time and take a look at what people were actually viewing. So when you build a dashboard, for example, you don't take the software and have the software run whatever dashboarding tool you're using. You don't have that dashboarding tool run the query to return a result set and parse it on the screen. What you do is the unified namespace runs the query, the unified namespace parses the data, and all the dashboarding tool does is visualize it.

42:51

Speaker 1

So I could take three different tools and I could plug them into the unified namespace and have them view the exact same result set that's only queried one time and been parsed one time. Moreover, the historian can store what was in that result set. So I can go back and look in time to see, well, what did they actually look at? Why did they run the wrong role? Was there some flaw in what we told them digitally that they should be running? The implications of the unified namespace, and this is the challenge in quote unquote, selling it is, the implications are far reaching. It's a philosophical and strategic decision to use a unified namespace. Right. Anyway, sorry, I know it was a long spiel there, but normally when we're doing this, we're sharing a screen.

43:42

Speaker 1

And so it can be, I keep trying to think, how is this going to translate as just great example, super powerful.

43:50

Speaker 2

I think what you just mentioned now is the key.

43:51

Speaker 3

Yeah, I think for me two things that stands out, Michael and Walker, is exactly that. Why must we in real time generate an oee number or account for oee? And then two weeks ago, or two weeks from that point in a report that's typical, our mEA solutions is that we've been doing all well for the past x amount of years. Now you need to run a query and reduplicate all of those calculations again in your query solution, your reporting solution to now have that same number. I love the idea historizing that already real time oee number that you've already calculated that's been produced on the namespace and you just relive it. I also love the idea of Joe. I mean, we've been looking at table reports now for how many donkeys of years? Right. I love your concept that.

44:42

Speaker 3

Why don't you look at the dashboard? The dashboard has so many other contextualized information, like you explained with what is the sizes of the whip, inventory of the wrapper. Why don't you look at the dashboard in the historical way? I think we need to get a little bit of a shift away from this old notion of static table reporting versus looking at what actually perspired, because you have all the context already available for you.

45:08

Speaker 2

Don't you want it in excel?

45:09

Speaker 3

No, definitely not.

45:14

Speaker 1

I want to say one other thing, or I want to make two other points that I talk about in other presentations all the time that we haven't really talked about in these podcasts. So there are basically two questions that will probably come up when people hear that story about that customer. And we do plan to demo the. I mean, there are other things where we're rebuilding their entire order system. We completely, fundamentally rebuilt the way that they manage art and they calculate their standards. There's a concept industry called the single pane of glass, right? Which is that all people interact with manufacturing and business data through a single pane of glass, a single entry point, and then they kind of navigate to where their specific data and information is. That's exactly the approach we've taken with this client.

46:09

Speaker 1

It's what we take with every client. But in order to be successful, there are two questions that are going to come up. Number one, how much does this cost? Right? How much does it cost to do this, and how do you accomplish it? What's the approach you take? The answer is that it costs way less than you think it does. So this

exact same client was looking at replacing just their ERP system, and they were going through, and they were getting quotes on ERP, and they had a separate company that was just going to do their dashboarding. And just the ERP alone, just the cost of the single ERP integration was greater than their total spend to this point, for all the solutions that we've developed around the unified namespace. Number one. Number two. Number two. And here's the most important piece.

47:00

Speaker 1

When they asked us, how much is this going to cost and how long it does it take? It was very important to say, we do not know. Okay, now, this is the mistake integrators make. Integrators try to come up with some number, and we all know that number is bullshit. When you're digitally transforming, you have no idea what it's going to take. Why? Because what they want is going to change based on how their knowledge expands. And what we've done. What we did was we said our gut feel based on what we know right now is on the bottom end it's going to cost you a quarter million dollars and at the high end it's going to cost you a half a million. You need to be prepared to spend a half a million, but we do not know.

47:43

Speaker 1

It could be a quarter million. Number two, we're going to do this one month at a time. How did this happen? We did it one month at a time. Each month was a phase or a sprint, however you want to describe it in agile. Each month we had deliverables. We delivered at the end of the month and each month we knew that were presenting for our future on the project. We were saying, you're going to get stuff every month and if you're happy with it, we keep going. If you're not happy, we pull the plug at any given time and we just did it in those very. And Michael, correct me if I'm wrong, we're on like phase 21 or something in the Mes system. And like phase five and order, job jacket. Where are the various phases?

48:31

Speaker 2

Those are 21 phases of incremental value.

48:34

Speaker 1

21 phases of incremental value.

48:37

Speaker 4

You got it?

48:38

Speaker 1

Yeah. That's not common in the industry, right? Yako, Lenny, you guys know this, right? It's not common to take that approach. It is not common to tell the customer, we don't know what it's going to cost because this is a strategy and what you want is going to change. If what you want is for us to quote you a fixed project where you're going to get a turnkey solution at the end, then don't use use somebody else. There are a lot of people who will do that for you, but you need to understand that you're going to pay double. There's going to be a ton of contingency built into that quote to account for the things that they don't know.

49:17

Speaker 1

If you're an integrator or you are a business leader, you have to understand that if you want to digitally transform and use a unified namespace, then you want to take incremental steps. Use this phased approach that we use, and by the way, we've applied it to every one of our projects now it's the one that works. And number two, you have to know, you do not know what your total spend is going to be, but what's going to change is you are going to get incremental value. Your time to value is very short. As each phase passes, you have stuff that is making your business better and your knowledge will expand and the new stuff that you want will be a function of that new knowledge.

49:58

Speaker 2

Yeah, 100%.

50:00

Speaker 3

Sorry, Walker, we see that a lot. I mean, we see that even if we just look about meas projects, not even talking about digital transformation projects, even if you try and run an MES project as a skater project, there is no deliverable at the end, right? It's not about I've ticked all the boxes, pack up my laptop bag and I'm off. It's about delivering that incremental value and expanding on what you see and what you get. So I fully agree, and I've seen too many MES projects fail directly of that reason. People think that it's something you deliver, you've got your set reports, you're done. And unfortunately it doesn't work like that. No, it doesn't.

50:39

Speaker 2

Great example, Michael and Walker. Thank you for that. When ready, I would imagine you're going to share something around that. When you're ready. I think that would be really valuable to our audience as well. Something I quickly want to just check in with you. We spoke about scaling and we spoke about adding capability. We spoke about simply dumping raw data into a data lake. We did speak about visibility, end to end visibility in terms of who can see what. On the topic of data interoperability and specifically, I don't know, I'm thinking of two examples where very often what we see is that there's custom scripts used to move data around from one place to another. We find that teams or people or individuals are spending a lot of time really just prepping and massaging data to get it ready in a usable state.

51:32

Speaker 2

So under that topic of, I don't know, what do we call it, data interoperability or data, what are some of the realities there around, what is available, what could easily be achieved, and maybe some tech that fits within that sort of bracket.

51:49

Speaker 1

So Michael, do you want to take a stab at this one first or you want me to go first?

51:55

Speaker 4

I can. So for the example, back to our use case here, the systems that they had in place, just to give you a feel for their technical staff, very bright people with an accounting background who learned how to program. Yes. When you look at some of the store procedures, they were running a standalone c sharp application on their production floor that was manually recording production data. When we opened the hood on that, it was clear that we had a lot of work to do around not only just getting back to what really is the raw data, but then also breaking down 20 plus c sharp classes and methods and all that, driving into their store procedures behind the scenes to see how they handle different business scenarios. And so what we did was leverage ignition for that.

53:13

Speaker 4

So a lot of times what you'll see is we would go back and source the raw data and bring it into an ignition data set tag. And then we would use expressions on top of that to map appropriate data set row and column values into a tag. So let's say it's back to the example of standards, right? We're running a work order for this customer X, and we finished a previous work order for customer Y. Now, behind the scenes in SQL Server, we're running a bunch of logic to understand what's the relative change over time. And so what we did in ignition was just bring over the raw standards. And as we see that, hey, we're transitioning from customer to customer, which was running product family x, and now we're on product family y in ignition.

54:23

Speaker 4

That's where we ran our business logic to basically recalculate what the appropriate setup standard, run standard, and everything else should be associated with that job. And we do that in twofold on their production scheduling tool and at runtime, when they actually start that work order one of their assets. So that's an example of where you have complex business logic and then leveraging a tool like ignition and expression capability, also Python scripting, et cetera, to basically extrapolate what that is.

55:05

Speaker 1

And one of the things I want to point out is, so Michael leads a team of developers, three people on his only I review their code just once a week. So on Friday evenings, I go into, we use a tool called fabricator to manage all of our code. A lot of people use Jira and I go through and I review, I just do a architectural peer review each week, and I never have to change anything on their team, ever. Michael will tell you, I never send them a message, never comment, hey, you need to do it this way, or hey, let's change it to that. And here's why. One of the challenges we talk about scalability here when we evaluate developers. So a lot of customers will hire us to evaluate another integrator, for example.

55:56

Speaker 1

They'll bring us behind, do a peer review, that kind of thing. One of the things that we're looking for is what type of coding habits do they have? Right? So are they the type of people, are they running a lot of scripts on the actual client itself. So that is when a client is fired. I open up ten instances, am I running the script on the client? Am I running the query from the client? Am I doing some function from the client itself when I don't need to? So in order to scale, what we have to do is we need to centralize operations as much as we possibly can. And by operations, I mean we need to centralize events, transactions. We have to centralize them as much as we can. Only query a database one time instead of 20. Right.

56:45

Speaker 1

Run a stored procedure instead of running a series of queries on SQL client side. Right, that kind of thing. The same thing with the namespace. Bring the data into the namespace one time, don't repeat it multiple times. So Michael's team is very good at centralizing all of their scripts. So all the python code gets centralized into packages and functions and methods. They're very good at using classes and instances. And what I want to get to here is this point. If you're going to be involved in digital transformation, again, I'm an integrator, or I'm a plant manager out there or an executive, your team is not going to be made up of engineers. Your team is going to be made up of engineers, project management professionals, and software developers. And you need software developers who you can teach engineering to.

57:39

Speaker 1

And you need engineers who you can teach software development to. And what you want are the experts in both fields on your team. Software developers are the ones who yell at the engineers for putting what should be in a class into an individual variable or function running on a screen. Right. That's bad software development practice, and it won't scale. Right. And it's engineers who yell at software developers for not doing appropriate levels of planning before they start writing their code. Right. Software developers will just jump in, and without really wireframing or actually writing down a problem statement, they won't do that.

58:17

Speaker 2

Engineers do.

58:19

Speaker 1

Yeah, right. What's important here is that the way you build your team is also changing. As an integrator. Most integrators have one business intelligence guy, if they have any. Your team should be made up. Half your team should have experience in business intelligence, not just one person out of 40 engineers, 20 of them should have experience in business intelligence. And if you've got 40 engineers on your staff and you don't call 20 of them subject matter experts in software development, you have the wrong team. You won't be able to do digital transformation. You're not capable of doing it. And so many things about the approach that you take has to change, right from the end user, from the customer's point of view, from the OEM, from the integrator, from the individual engineers and software developers. It's a completely separate way of thinking.

59:12

Speaker 1

And that's the reason that this has been such a slow progress. It's been a slog industry specifically. And it's very important to understand that scalability, you can build lots of great applications and you can build them wrong. You can build amazing capability and build it wrong, and it won't matter right up until the point you try to scale. And the second you try to scale it up, you're going to hit critical mass and everything comes crashing down. And

the last thing in the world you want to do is go back and refactor a project that's built on 250,000 topics inside of an MQTT broker.

59:52

Speaker 3

Yeah, 100%. And then obviously, it's normally the software that gets the wrap. All right.

01:00:00

Speaker 2

Software wrote itself.

01:00:04

Speaker 1

The platform is what gets blamed initially. With ignition, this happened all the time, and ignition got blamed. It doesn't scale. It's too slow. It's this. No, it was integrated incorrectly with ignition. The ignition platform specifically. I used to say this all the time. I still do. At its base level, ignition is as easy as it gets. Drag and drop. Put a tag in, drop it on a screen, visualize it, you're good to go. It couldn't possibly be easier at the high end. It's more powerful than any platform ever created in all of human history. And there's no problem you can't solve with ignition.

01:00:49

Speaker 2

Yeah.

01:00:50

Speaker 1

And I want you to prove me wrong.

01:00:53

Speaker 2

I think it's one of the points that we've at least heard over the last couple of months is the ability to do so many different things in so many different ways.

01:01:00

Speaker 1

Correct. And with that is great power. Right. Has unbelievable power. And with unbelievable power comes unbelievable responsibility. And it is very easy for incredibly gifted engineers to make a design mistake that kills them when they try to scale. Which is the reason that you have to approach these projects using teams of subject matter experts. You have to have peer review built into the process, because even the best engineer, even the best software developer is going to make a strategic decision that's wrong, that won't scale. Wrong means it won't.

01:01:40

Speaker 2

Yeah, yeah, absolutely.

01:01:43

Speaker 3

Walk. I know we're running out of time a bit, but I want to just touch on two things. I don't think we have to talk about it too much, but I do just want to iterate that. And maybe you can just confirm that people also shouldn't think that with any platform you'd be able to construct this unified namespace concept.

01:02:00

Speaker 1

Correct? I get this question a lot. Okay, can I use x for the unified namespace? Can I use y? Sometimes the answer is no, you can't use x. Sometimes the answer is yes, you can use x, but you shouldn't. And sometimes the answer is yes. Okay, right now, if somebody wants to ask me, hey, what should we use for our unified namespace? I think Michael will agree with me here and myself, Michael and Matt, who are the three guys who, Matt isn't a solutions architect yet, but he's architecting some smaller solutions. And we have a case study on a

project that he's led that I think a lot of people are going to be incredibly impressed with, actually. I know they will be. Our unified namespace now is, 100% of the time is built on MQTT, number one.

01:02:55

Speaker 1

Number two, the unified namespace now is being run through. I think EMQ is our broker of choice. Right, Michael? Have we settled that EMQ is the route we want to go? So we're using EMQ as a standalone MQTT broker, as the unified namespace. But then we are using solutions like ignition, flow, Canary, factory studio. Those are our four core application solutions that we are using to convert the raw unified namespace that's running an EMQ into something that people can work with, into tags and to a structure and events that they can work with. We're using aws in the cloud. I don't think we have any scenario where we're using azure right now, but it is very important to understand that OsiPi can act as your unified namespace, but it shouldn't. Okay, you could technically do it, but you absolutely should not.

01:03:59

Speaker 1

And this is the number one. One that pops up is, can I use osipie as my unified namespace? And the answer really is no. Why? Because a, it doesn't act as a broker, and b, it doesn't use MQTT to publish out. So it can only subscribe over MQTT. So you can't take the namespace that's inside of OsiPi and send it out. It also isn't versatile enough. Right. Our unified namespace of choice has been ignition. Over the last five, six years, it's been ignition. We've used factory studio in applications where speed is of the essence. So factory studio is much faster than ignition is. And because it's net, it sits right, basically right on top of the motherboard, whereas opposed ignition is wrapped in a Java wrapper. Right. So it's a little slower, but we use factory studio where speed is of the essence.

01:04:59

Speaker 1

But really those are the two that we use. We use flow as a node application which could in theory act as unified namespace with a third party namespace sitting next to it. So no, the answer is no. Not everything can be a unified namespace. If you're not a technical person and you haven't done this before, what you need to focus on are four bullet points. Number one, is it report by exception? Number two, is it edge driven? Number three, is it lightweight? Number four, is it open architecture? If you stay committed to those four things, you're going to end up at the MQTT three, one, MQTT five, or spark plug b destination. Nearly every time. Nearly every time.

01:05:43

Speaker 3

Thanks. That's exactly where I wanted to end off. And maybe just the last thing about report of exception is that exact timestamp we spoke about a little bit right at the beginning about timestamping and polling and exactly when events happen and report by exception is exactly what you need to get that exact timestamp, especially on the edge on when these events fire on your plant floor.

01:06:04

Speaker 1

And what we do, I want to say r1 quick thing, we're getting ready to do a big machine learning implementation for this client we talked about actually, and we're using the unified namespace to do it. And it's actually going to be a very easy integration. What we're going to be doing is sending two timestamps. We're going to be sending the event timestamp. That is the transition that was pushed when the event happened. Okay. And then we're going to be using a normalized timestamp for machine learning. And we're going to send both. So you're going to have, in the JSON, you're going to have a timestamp that we're telling whatever machine learning tool we're using to use so that we can get the rising edges lined up right. And otherwise it's fuzzy. Right. Machine learning people know what I'm talking about here.

01:06:55

Speaker 1

When I say the data becomes fuzzy. If I'm dealing with five data points and I'm looking for a relationship, one of the first things you want is when this thing happened, what was the value of this other thing? Well, the truth is, if I don't have the timestamps lined up, I don't know, it's fuzzy. Right. When was the rising edge of that following event or relationship event. So we're actually going to be publishing both.

01:07:21

Speaker 4
Both.

01:07:21

Speaker 1

The event that was pushed when it was reported, the timestamp and the normalized timestamp, to say to machine learning, hey, guys, these values are related. You can trust these normalized values. And that's a new thing, by the way. Not a lot of people are doing that. And it's something we're going to start pushing. Hey, start doing this in machine learning, you'll have a lot less problems. Send them both.

01:07:47

Speaker 3

Sounds to me like we've got a topic for session three.

01:07:52

Speaker 2

Outside of politics.

01:07:59

Speaker 1

I want to say thanks to Michael for jumping on, because we didn't prep for this. And I just kind of asked him, hey, man, we'll be doing this unified namespace thing. Are you interested in jumping on? Love it. And there was no prep at all. So I appreciate him coming on without any chance to. He didn't know what he was going to be.

01:08:19

Speaker 3

No problem.

01:08:20

Speaker 1

So I appreciate it, brother.

01:08:21

Speaker 2

Absolutely. I think it was massively insightful getting that. Michael, thank you very much for your time. Walker, thank you. Always amazing speaking to you. Always good catching up. It feels like we can almost do this weekly if we had the time. Love the passion, love the innovation, love the wanting to do more and do it smarter. Keep up the great work. Is there anything. I've made a couple of notes about Hibike specifically?

01:08:47

Speaker 3

I mean, if there's anything else to.

01:08:49

Speaker 2

Be valuable to share with the notes of the recording, please send that over.

01:08:53

Speaker 1

Yeah, what I do is I think I'm going to send an invite to the discord. If they want to ask more questions, or they want to talk to an entire community of. I think we've got 700 people in there now that they're not just intellic people. There are other integrators in there. There are Oems. You guys are. I don't know who. I think Clarice is in there. I'm not sure from your guys'team who's in the discord, but Arlen Nipper and Benson Hegland from opto 22, everyone's in there. So it's like a single place you can go and ask questions and get answered from Oems and from integrators and from architects and just machine learning professionals, if you want. That's really the place I think people should go if they've got more questions. It's a no sell zone, right?

01:09:49

Speaker 1

We don't let business development people sell in there. If somebody asks a question about product, if they say,

oh, hey, can somebody send me pricing on such and such? All of that's fine, but no unsolicited business development. I highly recommend anybody who wants to know more to join the discord. You can watch, obviously, our YouTube videos and LinkedIn, but we try to get the information out as much as we can. We're going to be doing a case study on this specific facility. We're going to be doing demos of the facility. They've given us permission to share the plant as know. We get to treat it as if it's our plant, right? So it's cool. I think it's going to really.

01:10:28

Speaker 2

What an amazing opportunity. That doesn't happen often. Yeah, that would be valuable. Thank you so much. Please do share that. I think all of us, we're never the holders of all knowledge, so I think the more learning you can do, we would definitely be a lot of folks interested in participating on that. Thank you very much.

01:10:46

Speaker 1

And by the way, I appreciate the fact that you guys are sticking with us. I absolutely love your podcast. I do. And I don't listen to many, but I listen to your podcast religiously. I really do appreciate that you guys are doing this. I know a ton of work goes into it and we need more of this. And you guys are doing it right. You know what I mean? This isn't marketing hidden behind a podcast. This is a podcast meant to really help. And that means a lot.

01:11:20

Speaker 3

That's our goal. It's really about Walker, I think you feel the same as me. I love innovation in this industry that we love. I love new concepts. And I think really it's for us, especially in the south african context, is just to get that little bit of innovation back. Playing a small part and playing a small part and get people to actually try stuff and try new things. It really is all that this is about.

01:11:45

Speaker 2

Thank you for the support. We definitely listen to all your stuff as well, but thank you for your support from over the pond.

01:11:51

Speaker 1

Appreciate it, brother.

01:11:52

Speaker 2

Great. Lovely chatting to you again. If there's anything outside of the link, anything else you want to send across, we can share in the notes. We'll do that. But thank you so much again for giving us your time on a Monday. Really appreciate it. Thank you, Walker. And thank you Michael as well. Great chatting to you.

01:12:08

Speaker 1

Awesome pleasure. Thank you, fellas.

01:12:10

Speaker 2

Thanks, everybody.

01:12:11

Speaker 3

Cool.

01:12:11

Speaker 2

So next week, Lenny, we have one more podcast left for this year, for 2020, and we call it a wrap.

01:12:16

Speaker 3
Yeah, it's our Christmas special.

01:12:18

Speaker 2
Are we singing? We can do a Santa, we can maybe wear a bright red jersey with.

01:12:24

Speaker 3
Luckily some reindeer on it.

01:12:26

Speaker 2
It won't be a video. We won't do any singing either. So we've got one more podcast left. I think we've lined up Michelle. We just have to confirm that she's been an integrator in our industry for, jeez, I think, 20 plus 20 odd years.

01:12:40

Speaker 3
Yes.

01:12:40

Speaker 2
And we're going to chat to her a little bit about how things have changed and also have stayed the same.

01:12:46

Speaker 3
Yes.

01:12:47

Speaker 2
So that's going to be a very philosophical discussion with Michelle, and we'll hopefully line up for next week. And that'll be our final podcast for the year. That'll be a wrap for the year.

01:12:57

Speaker 3
Yes. So we are obviously going to finalize our topics for when we get back January. So please, guys, get the topics and suggestions to podcast at Elementa.co.za. And then if we don't see or speak again, see you in the new year.

01:13:14

Speaker 2
Awesome. Thanks, everyone. Look after each other and stay safe. Bye.