Challenges Facing the Transportation and Logistics Industry — And How MuleSoft Can Solve Them

Mike Slack (00:00)

Hi, my name is Mike Slack. I work for AVIO Consulting. And today, together with MuleSoft, we're bringing you a webinar all about the challenges facing companies in the transportation and logistics space and how MuleSoft can help address those challenges. We're very excited, excited to bring this session to you. We're going to talk about some interesting topics and then take questions at the end. Today, the team that's bringing you this content is myself as well as Jeff Bean, a lead solution engineer at MuleSoft. There are about four topics today we're going to walk through. First, Jeff will take the lead in addressing common challenges in this space. Second, I'll talk about two of our clients and projects that we worked on, one related to IoT and the second related to three PL and third-party logistics processes. Both stories are really around how we can help scale those use cases and those initiatives. And then third, we'll address your questions at the end. I want to make sure and call out one thing. This is the Zoom-based webinar. You may be familiar with this technology, but if not, there's a panel at the bottom of this window that has a Q and a feature at any time during the webinar.

Mike Slack (01:08)

We encourage you to ask your questions. We might be able to answer them in real-time during the presentation, but if not, those would be questions that we can be prepared to answer as we get to the end. So we certainly encourage you to ask questions, be interactive. We look forward to that. And without further delay, I'm going to hand it off to Jeff to get us started. Thank you.

Jeff Bean (01:29)

Well, I appreciate it, Mike. Well, as we all know, in this all digital work from anywhere world, it's never been more important to sense and respond to changing market dynamics with speed, agility, and efficiency. The urgency of today's digital imperative is only increasing because people expect seamless digital experiences that meet them where they are, no matter the interaction. To make this real, 58% of customer interactions globally became digital when the pandemic hit. That's, up from 36% the prior year. And it's not going back.

Even when the pandemic is over, this trend with an increased emphasis on digital experiences will persist. And at MuleSoft, we're helping customers like Redwood Logistics go digital faster by turning every asset in their organization, every piece of data, every process, and every application into reusable building blocks. These building blocks are the foundation of Composability. Composability means using these standardized reusable building blocks like APIs to build new experiences, products, and services with greater speed, agility, and efficiency instead of having to write custom code or start from scratch every single time. This lets you solve pressing challenges today while simultaneously laying a foundation for the future, creating flywheel reuse that enables you to build faster with each project, accelerating your time to value.

Jeff Bean (03:07)

And you can do all this while maintaining security, governance and reducing technical debt. Consider this the demand for digital services and tools is heating up within and outside of It. The overwhelming majority of business teams say they need easy access to data and It capabilities to be productive as new digital projects increase with nearly three and five agreeing that integration challenges hinder business growth and revenue, but only 37% of organizations say they have the technology and skills to keep pace with those digital projects. What's more, It teams are being asked by the business to complete 30% more projects this year. This results in a widening transformation delivery gap that must be closed. Access to the completeness of data is critical and integration is all too often the main reason new initiatives fail, slowing you down when you need to go faster and widening that delivery gap between you and your customer and between you and the digital projects you need to deliver to make this real. Our research shows that integration

challenges are slowing down digital initiatives for 87% of organizations and it's no secret that It capacity is over subscribed but you can't hire enough people to do everything you need at the speed and the scale the business and your customers require.

Jeff Bean (04:46)

This is a similar challenge that Redwood Logistics faced when they came to Mule Soft. They wanted to move beyond a service provider to also being a solutions provider.

And they saw an opportunity in the mid market segment, which is traditionally underserved because of the cost economics. Yet it's a core growth engine in the US economy, creating

more than 90% of post recession jobs. One of the reasons why the cost of economics have not worked out, especially for Redwood Logistics, is new app development and feature development.
Took them two months to deploy their customer onboarding.

On top of that was taking six to nine months. That means six to nine months of missing out on revenue. And they're facing similar challenges everyone else is facing in this space. So how did they leverage Millsoft to overcome these challenges?

Well, they became a composable enterprise.

Organizations of the future will be composable when they built this way. They created what we call an application network. And in an application network, you have all the assets, business functionality and data that sits behind it in a format that allows you to know where it is, know what it is, and be able to securely access it and compose it. Just like a network effect, each building block adds exponentially increasing value and is strengthened as you unlock your data and business capabilities. Each note is secured by design and is ready for discoverability and self

service. The result is that Redwood accelerated the speed with which they delivered every subsequent project. These benefits are compounded as reusability drives a flywheel effect. Your ease and speed of composability accelerates over time as more and more assets can be discovered and reused. You can leverage those reusable assets to compose new products, services or capabilities, unlocking new business models and revenue streams. Composability closes the transformation delivery gap by improving your speed, agility and efficiency as a composable business. By reassembling reusable building blocks of packaged business capabilities like APIs, you go faster and faster. Project by project, you're more efficient because you're reusing not rebuilding. You can create, manage, monitor, and govern all of these assets at scale on one platform, and you can start to automate common processes and APIs. This accelerates your pace of innovation and time to value. Not only does this make you more adaptable, it puts you in a better position to scale your delivery capacity, support more initiatives, deliver projects faster, and drive business outcomes. Using this approach, you can deliver projects three times faster with reuse versus custom code. So let's talk more about how this is done.

Traditionally, integrations are done point to point, which typically aren't a big issue if you only have one back end system, right? You just only have to know one protocol, one authentication, any orchestration and transformation with that back end system. But as you start adding additional systems, that landscape starts getting more and more complex and backend domain experts become bottlenecks, right?

And you're now managing multiple protocols, multiple orchestration and transformation formats. With the MuleSoft approach, we take an API led approach where we start off with at the system layer for example the TMS system rather than connecting directly to that, we would create an abstracted Layer on top of that.

So now if someone needs to get shipment information rather than having to go to the TMS domain expert and hope that he's got availability within the next eek or two, you can actually access that information in some standardized formats. You now have a standard protocol, standard authentication and you don't have to bother that domain expert and you can unlock this information and data right away.

Another thing these abstractive layers do is if you needed to change that PMS System or any other back-end system rather than breaking hundreds of point to point integrations, you now only have to update one integration point and now all the other integrations are updated and working properly. On top of that we have our process layer. This is where you do the orchestration. This is where you need customer information. You don't want to have to know that that customer information is in 2345 different systems. You're just, like, give me customer information I don't care where it's at. This is where you can normalize and standardize what a customer looks like and provide that to that end user. On top of that, we have our experience layer. This is where you can create APIs that are specific to, let's say, a web app or a mobile device, right? Mobile devices need data in a different format. They don't need as much data as a web application does. So this is where you would create those experience APIs that are specific to that use case. Now, this is just our approach to integration and we leverage our Any Point platform to do this. Our Any Point Platform is the foundation of our product portfolio, providing a single unified system that combines integration, API and automation, design the management of those security and discoverability anywhere whether that's on premise, in the cloud, or a hybrid of the two. And now, with Composer, our newest product, you can empower business teams with a fast, easy and secure way to connect data and apps. We supplement our product offerings

with Catalyst, our unique and proven approach to customer success that enables you to drive business outcomes while avoiding common pitfalls.

With Catalysts, we bring together our proven methodology, technical assets and best practices to maximize your impact. All this is supported by MuleSoft's vast

practitioner ecosystem. These are the architects, the developers, the SaaS admins who gained expertise through our instructor led and self paced training, became certified, and who continue to help and support each other today. And as you can see at the top, security is enforced throughout everything we do. We provide layers of protection to ensure data is safeguarded at all times. Now, these are just some of the capabilities and reasons Redwood took this approach and went with Mule Soft to achieve the outcomes and overcome the challenges they were facing. Like I said, they saw an opportunity in mid market and they were able to capitalize on that by providing enterprise grade capabilities. They were able to provide those capabilities by decreasing their app development time from two months to two weeks. So they were able to be very agile and easily pivot with the uncertain and ever changing times that we saw over the last few years. And then on top of that, their customer onboarding was decreased from six to nine months down to 60 days.

That reduced their time to revenue significantly. And these are just some of the benefits that Redwood and our customers receive today. For example, on average, Mule Soft customers are able to do faster development and deploy at a 57% increase. Their time to market is increased by 78% and they're able to reduce their maintenance costs and effort by 74%. This is just an average. So these aren't benchmarks. These are what you should expect with the Mule Soft platform. Now, with that said, I'm going to hand the rest of the time over to Mike so he can actually share how abio has helped MuleSoft customers achieve these results and more.

Mike Slack (13:42)

Thanks Jeff. Well, I'm excited to jump into this use case and talk through what we were able to do with our client using MuleSoft and addressing this IoT use case. So first I want to talk through the process overall that we're looking at today, which is really an ecommerce worker fulfillment use case. In order to do that, I want to

spend just a minute talking about the transportation management system and the steps that are needed to go through and the various systems that need to be integrated with in order to make one of these deliveries happen. First, an order is going to come to an ERP through an ecommerce system or from partners or a third party. There are various ways that the orders can come in. After that, the company needs to get quotes from the carriers who are actually going to deliver that shipment. Then the carrier is going to be select based on the criteria that's part of that order and the destination. Then they've got to schedule that carrier and finalize the details of the pickup. Now at this time, the accounting team is going to need to generate orders and bills for the carriers while the customer team sends tracking information to the customer and the tracking shipment is going to come throughout the transit process and provide updates. There's going to hopefully be automated notifications to all parties upon delivery and then these key data points are going to be consolidated at the end for reporting purposes so our IoT use case is really just going to focus on a few of these steps. We'll come back to the end and we'll see where we've been able to interact with this use case overall as we finish up this section. So there are some internal challenges faced by our client that made addressing these needs of collecting this data and having this process work seamlessly. And those internal challenges are there's a desire for that data to be tracked by multiple transponder types. Now the transponders are sitting on the trailers or the trucks and they're collecting much of this information that's needed in order for these processes to happen. And our client works with various partners, so they don't own all of the trucks and the trailers and that means they also don't own the corresponding hardware or software that's on those trucks and trailers. That makes it very difficult for them to have comprehensive real-time fleet tracking. Many of these systems involve legacy technologies, so there are oftentimes paper, document reporting, and inefficient manual processes that are going along with these reports. And internally also, our clients face emerging and sometimes evolving digital requirements, new applications that they needed to provide, mobile applications, web apps, and a whole wide variety of new use cases are being presented to them as they're trying to more quickly address these basic needs.

In addition to all those internal needs, the story gets even more challenging because there's a number of customers out there who are mobile-friendly, technology friendly companies that are trying to push into this space and having some success. And where these companies have a competitive advantage is oftentimes around their ability to deliver software and new technology to the marketplace. They have more experience, it's more native to them to build mobile applications. Applications with lots of friendly visualizations, strong sense of user experience, and real time notifications. So their competitors are

oftentimes very skilled at the very things that are new and emerging technologies and capabilities that these transportation logistics companies have been trying to build over the years.

So a little bit about just a quick summary and a graphic here to visualize what these technology requirements are that our customer had. They needed an omnichannel solution. It doesn't matter what software it's interacting with. It needs to be agnostic to that hardware or software. They needed real time data from their fleet that supports an architecture that's scalable, agile, traceable and accurate. And just a word or two here on what we mean by scalable. We mean scalable in at least a couple of ways. One, scalable in the sense that we've already said where you can add in additional new hardware and software types into your infrastructure that's already developed and built but also scalable in terms of handling higher and higher volumes of data that's pushing through these systems. We need a solution that's not just working for the needs of today but as the company continues to grow both naturally and through M and A activity we need to be able to handle those higher volumes of data that are flowing 3510 years from now as well. So what did our team do to help meet these technology requirements? Our team put in place an API-led architecture.

It starts with that foundation of integrating with your core systems. And once that foundation is in place and the systems are orchestrated and the processes are optimized then the first pilot project can take place. And that's what happened with a client. We brought on that first transponder type. They were able to receive that data, process that data through the process layer APIs connected to their core underlying systems with their system layer APIs and understand the value of this approach. After the business could see the value, then the next project was to add a new transponder type and they were able to reuse all of those underlying API assets. The time to value for each successive project with that went down substantially because of reuse. Now, while the primary needs for this project were internal this infrastructure is now opening up future monetization abilities to expand beyond the core capabilities by empowering the business to partner with other shipping giants during busy seasons like Thanksgiving and Christmas, for example, a package that could be shipped through Ups might actually be delivered by our customer which is possible only because of the APIs available for tracking orders, shipment, and delivery.

The customer doesn't know what's happening underneath this technology and where it's happening, but because the APIs are in place, all these systems are able to talk together seamlessly and make this happen. So a quick summary of the outcomes that our client was able to achieve. They're able to track data from over 750 trailers each day, with more than 500,000 trailer statistics tracked every day. The experience APIs have been able to connect to multiple transponder types, and each new transponder type is able to reuse all the underlying infrastructure. You're able to connect more quickly because those processing system layers are already there to orchestrate the data into a canonical data model. And the system API is to receive that and put it into the right place or access from the right place. Additionally, experienced APIs are able to deliver the right data at the right time to new customer facing applications and for all their tracking and visibility needs. So to return to this first graphic where we started, we're able to see that in this transportation management system process, we're able to facilitate carrier selection, tracking of the products that are being shipped, the automated notifications that are going to the customers, to the shippers, as well as the reporting that's needed for all parties.

So MuleSoft API-led approach is what facilitated all of this for our clients.

Mike Slack (21:10)

All right, now we'll jump into the next Use case, which is all about scaling a three PL process. Unlike the previous example, which was about building something net new, this example will be about re architecting an existing process for efficiency and scalability. I'll be talking about many of the points that Jeff addressed earlier related to the Mule Soft platform, how it accelerates development.

I'll work to explicitly call out how.

We've seen those aspects benefit us in this use case. So we're focusing on a three PL process, and specifically where these blue dotted lines are here in this three PL process. So for our client or for any customer who is engaged in a process like this, we know that you're receiving orders from all sorts of channels, including e commerce, and you're also getting products shipped to your three PL warehouses. And there could be one too, where there could be 100 warehouses. And then from those warehouses, the order needs to get delivered to the right warehouse, and then the information needs to be carried forward so it can have the right delivery methodology, either going to a home delivery, going to a local

depot, or maybe a main hub or interim hub. Now, what we're working on specifically in this Use case is an order coming in from an ecommerce site. Having that ecommerce site, that order get routed to the right warehouse and that's the specific stuff we're focusing on. And within that, what we're really going to focus on is how do you scale and add up new warehouses more quickly. How do you bring new warehouses online and increase your time to value? Because when your business grows, one of the ways that you have to grow in order to meet the demands of your business is by adding more warehouse capabilities. So how do you build the software to make sure that your orders are getting to the right warehouse and you're incorporating those new ones quickly? That's the question we're helping to address with this use case. Now there's an existing process in place and that process was what we would call legacy point to point. There was a series of custom code that was written with a number of individual points and hops that need to happen. And each time a new warehouse was brought online into this system, every single step of this point to point approach had to be rewritten, rewritten for each new warehouse that was added into their overall delivery system. They didn't have any unit tests. There was limited traceability and observability and as I was saying, it's a monolithic app and a monolithic process it would take two to three weeks to implement an integration for each new warehouse. And because there was limited testing, there was custom code. Each new integration actually added more technical debt to the overall It portfolio. And so, as Jeff was talking about earlier about the delivery gap, the demands on It continue to grow, but the delivery capabilities in terms of resources and personnel on budget are remaining fairly static or around the same. And so with this process, you're actually putting more work on yourself by using an inefficient point to point process.

So on the technical side, those are some of the pain points.

Now, from the business side, they had different kinds of pain points. From the business side, they found it was difficult to scale. Whether through mergers and acquisitions or just through geographic expansion, the business wanted to scale faster than it could support. And that's not just adding on the new warehouses, it's also the amount of orders processed per warehouse.

As we said a second ago, with their legacy approach, each new integration, each new warehouse was adding additional technical debt. And as you process more orders, you're getting more errors. With limited traceability and visibility made the problem even more

challenging. They're using a reactive versus proactive approach, hunting down errors when they happen instead of architecting from the beginning in a way that would reduce those errors. And the business really wants to bring a new system online in days instead of weeks. So can we do something in two to three days that it used to take us two to three weeks? So that's the solution that we helped design for this client instead of all of these individual steps. What we did is we put in your standard mule soft three layer architecture where an experience API was bringing in orders from your ecommerce sites. A process API and orders process API was processing and orchestrating those orders and system APIs were connecting to the FTP server to a web service, which is connecting to your end delivery organization, and then your database for storage. So how technically is this architecture supporting their goals for scalability?

So, if we look at the next slide, a big part of this is what's going to be within this orders process API. This is what Jeff talked about quite a bit. Those composable building blocks. How do we make APIs reusable?

Well, you do that by designing them in the right way. We put a generic Ramble definition in place for these APIs. We used Message Identifiers and Transaction IDs for better traceability. We use dynamic routing to the downstream system APIs based on the message type or content. We put in Avio's standard loggers to enhance that visibility overall. And you're able to use configuration instead of coding by using properties in Mill soft in the way we've designed this. And then the dynamic database queries enables the reuse of the same asset.

Mike Slack (27:06)

So what's the overall benefit?

The overall benefit is that when you scale the next warehouse, what you're able to do is instead of rearranging the entire stream, as we saw earlier, you need to simply build a new system API to bring in your new system and allow that to interact with your existing orders process API. So this is how you get to two to three days instead of two to three weeks. That's just a quick summary of our use case scaling of three PL process and how we can dramatically reduce the time to value for our clients.

With that, we'd love to take some of your questions and see what thoughts you have about this session and all this content. So, without further delay, we'd love to take some questions.