EPISODE 1256

[INTRODUCTION]

[00:00:00] JM: The Traveling Salesman Problem is a classic challenge of finding the shortest and most efficient route for a person to take given a list of destinations. This is one of many real-world optimization problems that companies encounter. How should they schedule product distribution or promote product bundles or define sales territories? The answers to these questions constantly changed because business environments constantly change.

The company Nextmv helps solve these problems with production ready commercial tools for solving optimization problems and simulating models with real company data. Their tool Hop encodes optimization strategies for dynamic environments. Hop can be deployed to routing, scheduling, and assignment problems in multiple industries like on demand delivery, ecommerce, and IT infrastructure management. Their tool Dash is a commercial grade simulation engine that provides an environment to AB test models online with real data.

In this episode, we talked to Carolyn Mooney, CEO at Nextmv. Carolyn was previously a Lead System Engineer at Grubhub and a Decision System Analyst at Zoomer before that. We discuss optimization problems throughout different industries, machine learning strategies, and go into detail about how Nextmv move helps companies become more profitable and efficient.

A few announcements before we get started. One, if you like Clubhouse, subscribe to the Club for Software Daily on Clubhouse. It's just software daily and we'll be doing some interesting Clubhouse sessions within the next few weeks. Two, if you are looking for a job, we are hiring a variety of roles. We're looking for a social media manager, we're looking for a graphic designer, and we're looking for writers. If you are interested in contributing content to Software Engineering Daily, or even if you're a podcaster, and you're curious about how to get involved, we are looking for people with interesting backgrounds who can contribute to Software Engineering Daily.

Again, mostly we're looking for social media help and design help. But if you're a writer or a podcaster, we'd also love to hear from you. You can send me an email with your resume, jeff@softwareengineeringdaily.com.

[INTERVIEW]

[00:02:15] JM: Carolyn, welcome to the show.

[00:02:17] CM: Thanks, Jeff. Thanks for having me.

[00:02:19] JM: You work on Nextmv and I'd like to get into what that is. But first, I'd like to talk about logistics in general. You and your your co-founder both worked at Grubhub and these food delivery companies have a lot of logistical problems. Can you give me some of the general logistical problems you have to solve in a food delivery business?

[00:02:43] CM: Absolutely. So, my background actually in systems engineering, so it's thinking about how you combine a lot of different actors in your system, and what pieces of automation you have to put in place to manage that really efficiently. So, when we were at Grubhub, we worked basically on all things fulfillment, so automating fulfillment at Grubhub. Basically, what that meant is everywhere in the lifecycle of an order, or even a lifecycle of the driver, there were automation pieces. So, there was forecasting and scheduling. That was how do you forecast demand? And then how do you pick the best blocks to schedule drivers in so that you cover that demand really efficiently?

In the meal delivery space, you're serving restaurants, right? So, you have to make sure you have enough drivers to give them reasonable SLAs in addition to giving good delivery times to your diners. So, that was really important. We also worked on automated dispatch. So, that's the whole space around the Ubers and Lyfts of the world. They're doing it in mobility. We were doing it for meal delivery, where you have, obviously a live set of drivers and you have a rolling set of you know diners that are coming into your system and saying, “I want an order.” Well, you need to figure out who goes with what and in what order do you actually service those pickups and drop offs, because you're stopping at restaurants and then obviously making deliveries after that.

And then a little bit higher than that it was an understanding of in real time, so we did this like management of supply and demand for the forecasting and scheduling portion. But then real world is never exactly as you planned it to be. So, there was also the aspect of real time supply demand management, which we call that kind of market management at Grubhub. And then finally, there was ETAs, like how do you decide, how and when to surface ETAs and at what points in the order lifecycle.

So, we kind of had our hands in all of those various problems under the broader kind of umbrella, which we call decision engineering. So, how to automate decisions, how to make them look like code and how to ship them efficiently in your stack.

[00:04:34] JM: Did you say how to make them look like code?

[00:04:37] CM: Yeah. So, a lot of kind of what we built Nextmv around was this concept around decision engineering, and specifically like kind of decision science. So, those are two kind of newish terms, I would say, that we started playing around with but a lot of times what you'll see is like these models or algorithms, right? They are something that you might go off and research and then you're trying to pull that all that rich mathematical research, a lot of times done by operations researchers or data scientists, and you're trying to pull that into your software stack.

Sometimes that doesn't look and feel like other parts and other components of your software stack. So, if you think about like what are good practices and software in general, maybe like repeatability, testability, scalability, things like that, those don't often get serviced in the algorithm space. So, you there's this kind of divide between, “Hey, I go off and I research a really good mathematical algorithm. But how do I put that? How do we make it look like the rest of my stack and make sure it's following coding best practices?”

[00:05:35] JM: So, this sounds like a very tough problem to build a general software solution for, if you're trying to have a piece of software that models the logistics of a food delivery business, but maybe can also be used for like a ride sharing business, or any of these kinds of businesses where there's a lot of actors involved, maybe a fulfillment business. I'm just curious, how you generalize that problem set to a software tool that can be useful for all these different use cases?

[00:06:13] CM: Sure. So, essentially, what we've done, and I would say, in general, we're building out what we call the decision stack. So, there's kind of layers to this decision stack. And the first layer, I would say, at the base, is really our core, and that core is both optimization and simulation. So, a very high-level optimization can be used to automate decisions and pick the best scenario from a bunch of different permitted scenarios for your business based on your own KPIs. And then simulation allows you to do efficient testing. So, on top of that, what we've built is a set of engines so we kind of think about this and layers. So, our set of engines is really problem specific formulations. So, they are algorithms that are common to maybe logistics, or algorithms are common to finance or algorithms are common to scheduling or something like that.

So, what we have there, for an example is like a VRP. So, that's a vehicle routing problem. A vehicle routing problem is common across any optimizer framework that you look at. I would say it's one of the most common that people are familiar with. But a vehicle routing problem can actually apply to many different applications. So, that basically vehicle routing problem says, I have a bunch of drivers or a bunch of vehicles, and I have a bunch of requests, and I need to service them in the most efficient way. That means batching them together, and assigning them to a driver and then routing them for the ordering of pickups and drop offs.

Obviously, that relates to Rideshare, because rideshare does that exact problem that also relates to things like meal delivery, it also relates to sourcing. If you think about, “Hey, I want to source maybe different agriculture and bring it back to a distribution center.” It also goes to package delivery. You can think of UPS, FedEx, anybody that's doing like ecommerce, delivery, that sort of thing. So, all these problems kind of have these general formulations and what we've done is we've put kind of like an abstraction layer over that, that says, you can use these formulations for a variety of applications and define your schema, define where you want to run, whether that's serverless, or command line, or Docker, or what have you, and allows you to ship those things really easily. So really, what we're trying to do is give common patterns across multiple optimization and simulation paradigms. If that makes sense.

[00:08:23] JM: It does. So, can you walk me through an example, let's say like a food delivery business. Tell me how and why a food delivery business would use your software?

[00:08:36] CM: Sure. So typically, when you get, I would say, when you start to work on these problems, what you realize is very quickly, the combinations or the the scenarios that you can possibly generate for one plan. So, a meal delivery service, is typically going to plan iteratively, over the course of a day, because they don't know the new demand that's coming in. You and me as users, we could go top on and order our own food whenever we want. So, in that sense, it's like a real time delivery problem. Typically, what you'll see is that over the course of a day, a meal delivery system will run, I don’t know, maybe every 30 seconds every 2 minutes, somewhere in that range and take in the current state of the world. So, for our systems, we define that state in just JSON format and that's pretty typical for these companies that are working in like microservice land, everything is talking JSON and JSON out.

So, by defining that states in JSON, we can actually then pass it into one of our engines. So, our engines, like I was saying, we have a vehicle routing problem engine, which is what I would use for that meal delivery space. And that is what we call our fleet engine. What that does is it allows you to set different constraints on your system and we think about constraints as ways your operations limits itself. So, that could be things like your drivers have, a vehicle capacity that they can't exceed at any time during their route or they have to you deliver certain stops or certain packages at a certain time. So, there's time windows associated. Or there's an ordering of things that have to happen. So, meal delivery, you have to pick up food before you can drop it off. So those stops have to be connected, and make sure that you order them in that order. The pickup happens before a drop off, so we call that precedence.

So, there are these different pieces of that system that you can use are off the shelf, kind of like constraints for and configure based on your data. So, the interesting part with customization is I've never seen two delivery providers in any sector treat their delivery the same way. What I mean by that is they think about different KPIs, they think about different constraints, and I think we've got different combinations of the two. So, what we allow that's a little bit different, I would say than other maybe routing APIs is we allow you to customize the KPIs that you care about minimizing or maximizing.

So, let's just say you think about your entire delivery world in dollars, like you have everything kind of like scoped out so you know exactly how much a mile costs you, you know exactly how much waiting costs, you know exactly how much all this cost you. You could actually make your value function for the optimizer in dollars, based on the KPIs that are inherent to your business. Really, what's happening in that fleet engine is that at the end of the day, it's taking that initial state that you have, and it's generating a whole bunch of different scenarios. Under the covers, what an optimizer really is, is a search mechanism. So, it's the ability to both generate new plans, and then search those plans really, really fast to find the best one in the time you have.

That's that's basically how somebody would use us is they would take like our fleets engine off the shelf, they would configure their specific JSON input to match into that fleet, that fleet structure, and then they would layer on whatever constraints are important to their business and whatever value function they care about.

[00:11:43] JM: How is this so – I still understand how this so generalizable, because if I'm trying to optimize my food delivery model, it seems like I'm going to need data from a really wide variety of sources. I would need to have data from my food delivery business in Austin, and my food delivery business in San Francisco and my food delivery business in New York, and there's going to be different optimizations that you might make, based on the city, right? You might want to have different plans for different cities. So, is there a necessity of having a set of sample data in order to to actually build these models and test them properly?

[00:12:25] CM: Definitely. So, one of the things that we focus on was the ability to one unit test your bottle. So, a lot of other optimization frameworks, you can't actually unit test some of these individual constraints easily. And so, by having the ability to kind of like unit tests, that makes it a lot easier to have confidence, like some of those are being satisfied.

The other thing that a few of our users are already doing is kind of working this into their CI framework. So, they'll have a set of inputs that are basically snapshots at some point in the day and some of their regions to test out as well. And so, we've seen that pattern be pretty successful for kind of building this out.

I would say on the generalizable side or abstracted side, one of the things that's really crucial here is the mechanics underlying our solver. So, Pop is our solver. It's our hybrid optimizer. The mechanics underlying that are really based off of state machines. So, what you're doing with like your specific input is you're defining your initial state, and that state has to represent your set of requests and your vehicles in the meal delivery case. But it doesn't necessitate having – basically, there's not a lot of specificity between Austin versus Denver. I would say the specificity comes in like maybe the travel times and stuff like that. But those you can use a variety of approximations. You could either use like straight line distance scale to some measure, or you could use map-based routes and stuff like that to pass in. It just depends on what you want to do.

[00:13:45] JM: Cool. So, let's talk a little bit about the engineering. So, give me an overview of the architecture, I guess. I mean, there are two products to talk about here. One is your Hop product, which is decision modeling and optimization, then you have another product called Dash for simulation and experimentation. I guess, let's actually talk about those two separately, and then we can dive into how they're actually constructed. So, decision modeling and optimization. Can you define that problem space first, and then go into simulation experimentation?

[00:14:24] CM: Sure. So, most of what I've been kind of talking about, I would say in the middle delivery world has been on the optimization side. So, when you're thinking about optimization, it is at the purest form, generating possibilities of what could happen based on the input data that you provide and picking the best one based on your defined criteria, which we call a value function. That value function, like I said, can be a combination of whatever your KPIs are, could be a single KPI, it can be total distance traveled for all the drivers across your network, for example, in our VRP. So, that's basically like the context of the space around optimization.

Optimization is quite a large bucket of things. If you Look at you know where optimization is applied. we're only talking about logistics right now. But even within logistics, it's dispatch. So obviously, the routing and scheduling or routing assignment problem we're talking about now it's workforce scheduling. So how do I pick my best blocks for those people? It's also your warehouse management. It's even in places around like we talked about market management.

There are just like many, many, many ways to slice and dice how to use optimization. But it's a fancy form of search at the end of the day.

How we’ve constructed that is, we're using an underlying technology called decision diagrams. So, there are multiple camps or paradigms of optimization. Decision diagrams, is actually relatively new in the academic sense. So, in the past, I would say like 10 or so years, a lot of research has been done out of Carnegie Mellon and elsewhere, around decision diagrams as a way to solve optimization problems. And so, we leveraged that research plus my co-founder’s own PhD research, to basically build the first commercial decision diagram solver. So, that was kind of the first modeling paradigm that happened with Hop.

Some other ones in the space, for listeners who are a little more familiar with optimization, or like mixed integer programming, or constraint programming, these are all different ways to do the same thing, which is just to automate a decision that is complex, essentially, where there's a lot of interdependent decisions that happen in that plan, and somehow produce various results of KPIs. I would say, classically, optimization really hits the bottom line of every company. So, this is something that is crucial in almost every sector, it's just a matter of like, what decision is being driven out of that optimization.

So, our tech stack is built entirely in Go. So, we actually built our solver, that is that core that I was talking about at the base layer, and everything is entirely built in Go. I kind of made that choice, largely for deployment reasons. So we actually, have the ability to compile these applications or algorithms down to binaries, that you can kind of put in some various contexts, whether you're running it locally command line, or whether you're running it in web app, whether you're running it in serverless, there are a lot of different ways to kind of deploy these binaries, it also gets to the component architecture. Because it's basically like, a micro service, it has a defined input and output structure. It's easier to manage in your software stacks, it's not kind of scattered across different micro services.

[00:17:22] JM: Can you give me a little bit more information about how you’ve selected your infrastructure? What, I guess programming languages, decisions, database decisions, cloud hosting decisions? Just take me through a little bit of how you frame your thinking for the company?

[00:17:39] CM: Sure. So, a lot of it is around thinking. So, I kind of mentioned this, like decision stack, the way we've thought about – so our cloud product is relatively new. Initially, when we started the company, we were thinking of ourselves more as like a library of tools for our users, more so than a hosted product. I would say over the last year or so, we've realized that, from a developer tooling perspective, it would be great to have a hosted solution. So, we've really thought about this in the sense that our cloud is an instance basically of our decision stack. The way we think about that is our decision stack involves a deployment mechanism. Like I said, that deployment mechanism can be a variety of things for our cloud infrastructure. Right now, I think we're using Lambda in the backend, so like Lambda set functions, stuff like that. So, that is just like one particular formulation of using that. But you could also do it in a variety of different ways, different cloud providers different aspects of that for whatever.

We really wanted to target smaller services on the on the cloud side and the reason for that is, typically optimization has been this thing, I would say, used by more legacy industries, because of the compute resources required, and also, just like the amount of data required. So, I think that this is changing. I think one, just the data handling across the industry has become a lot better. And also, the introduction of like a lot of like cloud resources, and scaling down of their cost, has really made this more accessible. The same actually goes on the simulation side, because simulation side, I was working at Lockheed Martin, and obviously, they're a giant corporation and had a lot of money to spend on private servers, that sort of stuff.

So, we made some decisions that way around trying to scale down this technology to be able to run on like on things like serverless. I think that's really been helpful in keeping us focused on performance from a software perspective. So, making that search for the optimization really, really fast, and really efficient in terms of picking good solutions and improving solutions over time.

Typically, with optimization, like, operationally, you don't really care about the mathematically optimal like, yes, mathematical optimal is great if you can get there, but a lot of times operators are okay with like a 5% within optimal solution if it gets back to them and like 10 seconds or something like that for a really complex problem. So, we've kind of gone more down that path with the software architecture. I know you touched on like databases and stuff like that. We haven't really had the need to go down that path super heavily yet. I would say that is something that we're basically leaning heavily into like the AWS stack for our own personal instance of our decisions stack. But most of our users right now are actually self-hosted. So, they view us basically as a collection of tools that they can use in a self-hosted capacity and build out their algorithms that way. Does that answer your question?

[00:20:32] JM: It does. Give me a sense of how compute intensive this kind of modeling is. If I've got a bunch of different permutations, I could potentially run to determine the optimal decision science across my company, it seems like that could be quite compute intensive. Correct me if I'm wrong, am I wrong about that?

[00:21:00] CM: No, you’re not. In in terms of like actual number of solutions, or like I would say, pieces of that tree that you're exploring, like, that can be anywhere in the thousands to millions of nodes in your search tree that you're trying to explore really quickly. So, we've come up with, I would say, like strategies for searching that space. So, different ways to manage it, we think about it, like that tree is like a really large diagram. So, there are ways that you can search portions of it really quickly, or dive to feasible solutions, and then kind of work your way from there.

So, there are a lot of like optimization, I would say, paradigms, or like optimization strategies to manage that space really efficiently. That kind of gets into the why we chose decision diagram structure as our optimization paradigm more generally. I would say from like, a compute resource perspective, the fact that you can run this on Lambda means like it's actually pretty sharp on that front. We're running like, thousands of deliveries through Like lambda functions, essentially, that have a cap out at, whatever, 15 minutes. So, from that perspective, they're able to run pretty efficiently there. And a lot of our users in the real time realm are getting results back in seconds or milliseconds. And that’s required for the type of planning they're doing.

So, verily, is very problem size dependent, as well as like complexity of the model itself. So, how many constraints are you considering? How many different combinations like all that kind of stuff, and some of these things are operationally limited, and some of them you kind of introduce as ways to make the problem like a little bit easier. In optimization tree, that's called relaxation of problems so that you can find good feasible solutions quickly.

[00:22:39] JM: How do you make it reliable that the model reflects the real world? What I mean by that is like, is there a workflow where these companies, the potential customer, like Go Puff, or Rappy, some of your customers, where they've got their test data, they've got their decision inputs, they do their modeling, they change how they run their business based on the model. But then they would want to verify that they're the sub positions of their model, are correct, right? They would want to have some workflow that they can verify that things are going as planned. And then if it's not, they want to iterate on it. Can you take me through the kind of life cycle of iteration that a company would use with this kind of modeling technique?

[00:23:36] CM: Sure, that kind of gets into the space around experimentation. So, I'll touch a little bit on simulation as we go here. But essentially, what that lifecycle looks like is, typically when you're building, I would say, decision automation into your company, it goes through a variety of phases. First things first, is usually like some sort of manual version of that, and we were part of the do things that don't scale like, that's pretty common. If you're doing some operational problem, you're going to do the manual version first.

Typically, what you see in a company after that is, they go into this realm where they say, "Hey, I want to automate that decision." That's usually when you'll either do one of two things, either build out some business rules as like the heuristic and just use that as part of your decision framework or get into the space around around optimization, which is what we've been talking about thus far. And so that would be the how you automate that, that singular decision. After that, you kind of start stepping up into what I call like the experimentation realm.

So, one is you can use simulation. Simulation basically adds some randomness into the execution of that plan. So, you may have your decision model or algorithm decide on the plan for you, and maybe that says, me, "Carolyn, I'm going to run to X, Y, Z locations and I expect to be there." The output from my model says, "I expect to be there in 10 minutes, 20 minutes, 30 minutes, et cetera." But then when you run it through simulation, you add some noise to my travel time, right? Because nothing in the real world actually happens in the way that you would make a plan. So, my ETA to that, first – or my actual arrival time to that first location in the sim world might be 8 minutes and the other one is 12, and the other one is, you know, whatever 34.

So, that allows you, basically, to understand kind of how sensitive your decision automation is to kind of real-world events and shift in that plan. So, you can run simulation and kind of out for maybe multiple hours or multiple days and kind of see how things back up in your system. Because at the end of the day, a lot of these logistics systems are basically built off of queues of some sort, queues of orders coming in, or queues of packages that have to be packed. stuff like that. That's like one layer, I would say, of testing.

Even before people get to the I would say, the simulated layer of testing, they typically are running like an instance, verse instance. So, if you think about, "Hey, I took my state of the world at 5 PM on a Friday, which is probably one of the most order, the highest", or sorry, highest number of order times, for a lot of meal delivery services. If I take like that 5 Pm on a Friday time slice, I can run it through my current algorithm, and I can run it through my new algorithm. And I can say, "Okay, what is the difference in my value function?" Because typically, I'm trying to capture the same value function, whether it's total distance, total time, lateness for ETAs, stuff like that. I can compare those two values.

So, that's like a first pass, I would say, at looking at acceptance for like a new model, and then the second pass would be simulation. Simulation really just allows you to narrow the field and what you're going to go test live. So, it should be able to give you directional information like this is probably better than then this other algorithm, so you should go test it, real world. And then when you step down into the real-world context, you might do what we call the switch back testing. So, for certain slices of time, have one algorithm live versus another. You can't do traditional AB testing in the sense that you can't split your driver pool in your meal delivery world or in like rideshare world, because then you're fundamentally not getting an optimal solution by not considering the whole pool together in one planning session. So, that's kind of typically what you would see in terms of experimentation, you would kind of step through these different phases, and have some acceptance criteria throughout.

And then on the end of it, you're looking at, basically what is the actual result and compare back to your model. A lot of these providers, you know, think about travel time, I would say that's a definitely a source of error in a lot of models that you typically have to make some judgment calls on. Because you can't be querying Google Maps for every single possible connection, in every single possible part of your plan as you go through an optimization. So typically, you'll see either passing in different travel time matrices, and those are dependent on a certain time of day from like any travel provider, or travel time provider, whether it's OSRM, or I think one of them is Valhalla, GraphHopper. There's quite a few out there.

But that's one thing you could do or you could also build a regression model based on your observations of travel time data, because a lot of these services are observing all their drivers moving around all the time. So, they can kind of incorporate their own data science models into the input for optimization. So, that's kind of the nice way, like decision science is kind of this missing link between data science, which is all about, like the prediction of what's going to happen, and then the operation is like, “Okay, what do I actually do about it?” And in the middle there, is decision science, which gives you a set of actions that thinks is going to give you the best outcome.

[00:28:25] JM: How generalizable is the technology that you've built? I think we've mostly been talking about this in terms of logistics and kind of these these real world decision science problems, but on your website, you also have the you have suggestions that it could be used for scheduling jobs in the cloud, building marketing engines, marketing engines have have all these different flows to them, like you want to target a person with an ad on one platform and you target them on another platform. Take me through how generalizable this is and and how are you planning to approach these different markets?

[00:29:08] CM: Sure. So, like I said, with the core technology there, is based on like, say, machine mechanics, which are, I would say not specific to any field. And you look at something like maybe a marketing engine, for example, that might be looking at, “Hey, what are my different channels that I can go make marketing decisions in and allocate resources to? Whether they'd be fun or something like that.” You would typically have data science models that say, I expect this certain ROI on that particular channel. So, what you might do is that across all of your different users and all your different channels, you make an optimization decision that says like, find the best results in ROI given that I have to allocate as a finite amount of resources or dollars to those different channels. So, that would be another example of like an optimization engine or application that you can build within your company, and so that's kind of interesting.

Another one that’s kind of interesting is looking at the workforce scheduling, workforce scheduling, yeah, we talked about it for meal delivery logistics, but it's actually quite common. The scheduling paradigm goes into healthcare, on scheduling different providers for different blocks and patients coming in. It also applies to obviously, like ecommerce, retail, et cetera. So, they kind of go into those different areas.

The way we've been thinking about our platform is very much a horizontal play on giving optimization as part of your toolkit, or developers and then building applications on top of our core, that makes sense for certain industries, because I think one of the hard things about optimization is it is so abstracted, and you've kind of touched on this a few times where it's – but is it possible to be that abstracted, and to really be relevant? I think like, that gets into the having specific schemas and applications that we can launch in our cloud service, to give people an understanding about how they can use us. So, whether it's that marketing application, or whether it's the meal delivery application that we've been talking about most of the time, those are have a specific data structure, have a specific output and have like a very tangible decision point. I need to like assign drivers. I need to allocate dollars to my marketing channels. I need to pick my schedules for my healthcare workers. Those are very defined outputs.

So, I think that's one of the ways that we're looking at addressing different pieces of this market is kind of giving these applications use as is or customize as needed. So, that just gives people a sense for what they can use it for.

[00:31:41] JM: Who are the teams that are responsible for using decision science software within an organization?

[00:31:49] CM: So, it depends on the maturity of the company, or even just the style of the company, I would say. So, we have two different types of users, I would say. We have our typical software engineer user, that's the one that we've primarily targeted with our toolset as far. And the reasons for that is a lot of the optimization space hasn't really been that sharp on integration patterns. So, we've really tried to make a pretty big difference there and also in testing. I was talking about with the CI testing, and also like unit tests, that sort of thing. And then we also have data science users.

So, data science users are, I would say, the classic consumers of optimization technology. Typically, they will be operations, researchers, not just data scientists. So, just a little bit of a qualifying factor, because I know there's a lot of different types that fall into that data science title. But these operations researchers are typically like master's students or PhD students that that went through operations, research training, and they're out there looking how to solve these complex problems in industry and they will usually use a solver off the shelf. The reason for that is, a lot of times, you're doing so much work to define what the problem is for your specific use case, that building the solver itself is not really time well spent. Quite frankly, it's a complicated technology. It's kind of similar to like Elastic Search, like you wouldn't go off and build the Elastic Search instead of just using it because it's copied enough technology and it's not really specific to your business. You just want to utilize it as a toolset to solve your problem.

[00:33:22] JM: How do you see this domain evolving over time? I guess, take me inside your product strategy, and just how you see the industry evolving to have more requirements for this kind of software?

[00:33:36] CM: I think one thing that's been really interesting is that you've seen, or at least we've seen more and more of the like systems thinking operations, researchers, data science be part of the general software company. I think that's because of a couple things. One, data proliferation. There's data out there for almost every aspect of your business, whether you're getting like IoT signals, whether you're just getting better data as your users go through your funnel, stuff like that. I think that there's a lot left on the table for optimizing those flows from an operations perspective. I think, because you have access to that data, because there's more of this kind of like systems thinking around, “Hey, I have a marketplace and I have like two to three to four sided marketplaces now that all have these really like competing interests, on every side of that marketplace, and how do you really balance all those factors that you, the company that's facilitating this marketplace get the best benefit?” So, I think this space has kind of grown out of the fact that a lot of businesses today are not really transactional anymore. There are a lot more factors at play. There are factors at play that have data associated with them now that wasn't exposed because you can store all better because you just couldn't see it visibility perspective.

So, the digitization, I think, is really driving a lot of this change. I would say the second thing that's driving it is just around this gap between data science and operations as it exists today. Data science is really like amazing work on predictive modeling and understanding LTV and understanding all these things for various businesses. But at the end of the day, how much of that is actually crossing the bridge into the real time operations or the tactical operations of a company on a day to day basis? So, I think, the space around decision science is really taking action on all those rich predictions, all that rich data, and giving you the best solution in the time you have, for the time you have to make these decisions and then obviously, the monitoring, and what have you that comes out of that is part of that story as well.

[00:35:41] JM: What are the biggest technical challenges that you're working on right now?

[00:35:47] CM: Yeah, there are a couple, one of them is I would say, because we have these two users, we have our software engineer user and our ops researcher user who's typically like that masters or PhD. I think, one of the challenges that we have is usability from from both ends. I would say, understanding, so we've talked about building that app layer, it's just a knowledge thing, like we are out here, we're helping solve these problems, and you can take us off the shelf and use us for that software engineer.

So, from that perspective, I think is a visibility thing on the technical side, and just making it so that there's enough rich content around what we're doing and specificity in our applications, to get people to understand like the application. And then, I think on the other side of the house on like the more PhD user side of the house, one of the challenges is scaling. Scaling of these problems, as we get bigger and bigger, being able to run their problems in things like serverless, being able to like paralyze, like different parts of search. Being able to use different paradigms. We are adding heuristic paradigms to our optimization core. So, as we go with that, obviously, like, thinking through how to use both our current decision diagrams and meta heuristics and another optimization framework, like constraint programming, how do you make those look and feel the same from like an ergonomics usability perspective, and I think that's like a really interesting technical challenge for us.

[00:37:23] JM: I would love to know more about your perspective on the on-demand and like logistics, heavy economy in general since you've spent significant time in it, and many of your customers are in that space. We're at a point where you have so many different players that offer faster and faster delivery for all kinds of items and things are just becoming easier and easier. Is there something that happens next? Is there like a next phase of what ecommerce and logistics and the on-demand economy offers? Do you have any vision into what's kind of the state of the art is going to unlock?

[00:38:10] CM: Yeah. I mean, I think there's a couple interesting trends going on there. You see a lot more now, like shared fleet situations, or the white labeled fleet situations. I've seen from just my own usage of these kind of applications. I've seen basically, like, DoorDash, I think is like white labeling a lot of their fleet usage across like different restaurants and allowing them to kind of control the user interaction, which I think is really interesting. Typically, a lot of these players have kind of started out as ecommerce marketplaces, really, at the end of the day. And then they've tacked on fulfillment as like part of their service set. So, I think there will be an interesting – there are definitely some interesting things happening in that space, especially given the pandemic, honestly. There's been a lot more reliance from a restaurant perspective, in this space, on delivery, because they just can't have the same amount of volume that they used to have in their restaurants real time.

So, I think that that will start to shift a little bit, kind of like ownership of data is something that I'm super curious how that grows and develops over time, and kind of who owns the customer at the end of the day. I'm also interested to see how these like shared fleets and even how people start to use autonomy in this space because, a lot of – I think there's been a misconception that you have unlimited elasticity in your supply of drivers. I think what we've seen is that like gig economy workers, like there's a lot of competition for them now, whether it's ecommerce delivery, whether it's meal delivery, whether it's CPG delivery, whether it's rideshare, Lyft, Uber, et cetera. I think we're all kind of seeing that you're accessing, obviously like a section of workers that is interested in like ad hoc work, which is awesome. But I don't think they're unlimited.

So, I think what you're what you're seeing now, with extended fleets and partnerships on that side, and also different ways or whatever to like use autonomy is that ways to extend your fleet and have like more consistency in the supply level that you have. So, how do you start to see benefits on that side? So, I think that's going to be something that's kind of interesting. And also, just from my Lockheed Martin background, it's kind of interesting to me for a variety of reasons. So, that's kind of cool.

I think the other thing that's going to be – I think the industry in general has been driving towards faster and faster and faster delivery times. I do wonder if that will change or develop into more what is the expectation is of a user, because I think sometimes, at least, I know, personally, there are times when I'm ordering food, where I only care when it comes. I might pay a little bit less like to have it come later, or what have you. There are times where I'm like, “No, I really need to eat between X, Y, Z hours.” So, I think there's some consideration of like that, too, how do you how do you better serve users, and also gain as much efficiency into your logistics operation as you can?

So, I think those are two kind of interesting things. Yeah, I think the restaurant one is definitely concerning. I would say, individual restaurants, I don't know how much they can like really afford staying in the current paradigm, especially as in-person dining, really isn't coming back as quickly as anticipated.

[00:41:31] JM: As those those kinds of companies evolve, since you're already offering these kinds of modeling and experimentation solutions to them, and given that you've been inside of these companies are you've been inside of Grubhub, at least. Are there any other kinds of applications that you think they have, like a dire need for?

[00:41:57] CM: I mean, honestly, I would say like, they're all working on on these problems today. I would say it's just like, a lot of it is like the way it integrates with our business-like speed iterations, like ease of experimentation, like that sort of stuff is really, at the end of the day for one of these large providers part of the value proposition here. The other thing is, this makes a little bit more consistent. I think one of the things that's interesting about a Grubhub, or an Uber or Lyft, or whatever is, they're probably using different technology across all these different decision automations. So, there is a little bit of an aspect to resources being able to shift easily throughout your company and kind of use the same stack for decisions. So, that's something that we've been thinking a lot about whether it's your standard operating procedure for a very simple workflow to this routing decision that's really complicated. Wouldn't it be nice that they all look and feel the same, in the same way you use like Java for your whole stack or what have you? I think that's a really interesting thing that is more on the resource and innovation side of the house than it is, for this specific model, how do I compare? It's like more of like a force multiplier, I think, for the organization as a whole.

[00:43:12] JM: Tell me a little bit about your personal experience running the company thus far. What has been the most challenging and what has been easier than you expected?

[00:43:23] CM: That's an interesting question. I think one of the things that has been – I don't know if it was easier than expected, or kind of somewhat expected, but we ran a lot of the Grubhub team remotely. So, both Ryan and myself, my co-founder and I ran that team, and we got up to about like, maybe like 40 or so people running completely remote and he and I were not in an office at the time. So, I think like one of the nice things that we were kind of prepared for this, like pandemic environment that way, we had also intended to build our company distributed. I would say that that's been working out better even than I expected. I thought we did a lot of things, a lot of fun things like with our team at Grubhub, in the remote aspect, but we're completely distributed on this go around. There's nobody that is actually in an office for the Grubhub, since we had a couple people in Chicago, a couple people in New York, that sort of thing, and then everybody else was kind of at their home base. But I've been like pleasantly surprised that we're working – right now we're working across, I think four or five time zones, including Germany and Columbia. We have, I would say, a really high level of engagement and just a lot of like fun with the company even though we haven't been able to meet in person. So, I think that that's been really interesting.

On the flip side of that, I I wish we could have met in person already. I think that's been like one of the challenges that we just haven't had FaceTime literally. I haven't met half of our team in person, which is kind of crazy and that has its own challenges. I think we've made a lot of strides. I feel a sense of connection with every single employee that we have at Nextmv and I think that's awesome. But how much stronger would those bonds be if we were able to have our quarterly offsites like we planned? I think that aspect could be could be a lot easier.

I think one of the other challenges that I'm starting to get in here into the now is like, not knowing what we don't know, in terms of like community building and that sort of stuff. The external and community building, I think is an area of interest for me that we've been doing a lot of research in lately. That, I would say, not a negative, but it's just challenging. I think one of the things about a business, I spent a lot of time in analysis and engineering and now thinking about all aspects of the business, right? So, it's fundraising, and it's runway management, and it's budgeting and hiring, and also go to market strategy, all the things.

So, that's been fun, but challenging, and we've really leaned on a lot of our investors in that capacity. We were fortunate enough to have amazing investors in our seed round. We had first Mark follow on and lead our A. They've just all been super, super helpful and we've even brought out a bunch of founders and execs from companies like GitHub, and Twilio and stuff like that. So, they've all been just very, very helpful. I really think about our investors as like our strategy team, helping us think through things that we don't know, and we kind of specifically picked investors, advisors, et cetera, as subject matter experts in areas hopefully, that we didn't have as strong of ties to. So, that's been really helpful.

[00:46:30] JM: You mentioned before the impact of the pandemic, on your business and software companies in general. Any other unexpected ways in which the pandemic has affected business and maybe anticipations of long-range impacts there?

[00:46:48] CM: Yeah. I think, one of the interesting things for us is that we thought about our go to market in the logistics category, just given one, the application of optimization there, and two, our backgrounds having some cloud in our area. When the pandemic happened, I think everyone was kind of like, “Okay, well, what the hell is going to happen if I go to market? And what's going to happen to my customer base?” What happened in our customer base was that their services just exploded with growth.

So, if anything, our customer base kind of felt the pain and need of this technology, even more so than normal, because they're not only were growing rapidly, and so, fundamentally, the way they thought about their optimization was changing, or the way they thought about automating their systems was changing, but their operations also changed significantly. There were new rules that had to be abided by, because of the pandemic and there are changes in their operations along the way that, honestly, now that we're a year plus, and I think might hold. So, they've had to think through ways to incorporate those changes into their automation and iterate on their models, and like, all this stuff. So, in a weird way, I think pandemic actually accelerated our growth and that was part of the reason that we got interest from seed to series A was only about like six or seven months for us.

So, part of that is just like the acceleration there and the other part of that is really the democratization play. So, I think just more and more people are realizing that they need to manage these systems that way and it's just highlighting how fragile things can be if you build them off of like just a very clear cut like set of rules. I mean, you don't consider a lot of options. You saw the logistics industry in general was like really taxed, especially at the beginning of the pandemic. And so, it was challenging a lot of like old assumptions and old models and old technology that was being used. So, in that way, it's been like a really ripe time for us to step into the market.

[END]