EPISODE 871

[INTRODUCTION]

[0:00:00.3] JM: Data engineering allows a company to take advantage of the large quantities of data that the company has generated. In many companies, new data has been produced rapidly for many years, but the company has not been able to take full advantage of it. Creating large data sets does not provide immediate value for a company. A company needs to perform data engineering and data science to take full advantage of that data.

When data gets generated, it is stored in a database, a data lake, or an API backend, like Google Analytics. In order to manipulate that data, it is often pulled into a data warehouse. A data warehouse provides fast access time to large quantities of data. Pulling data from a source, like a database, or a data lake into a data warehouse requires a process known as extract and load. Once the data is in the data warehouse, it may also undergo a transform which enriches the data and puts it in a format that is easier to make use of.

Once data is in a data warehouse, it can be used to build models, interactive dashboards and Jupyter Notebooks. The data engineering lifecycle has many different components, which is why data engineering can often be intimidating to a company that is trying to make use of their data.

MELTANO is a project with the goal of providing a system of conventions for managing the data engineering lifecycle. MELTANO was started by GitLab. The MELTANO project has some strategic similarities to GitLab. Danielle Morrill is the General Manager of MELTANO at GitLab. She joins the show to discuss the world of data engineering and the architecture of MELTANO. We touch on the different components of a data engineering pipeline and the most acute pain points for data engineers.

If you're looking for all of our episodes on data engineering, check out the Software Engineering Daily apps for iOS or Android. We just refactored the Android app. Both the apps are in good shape. You can find all the episodes there. You can find episodes about Kafka, or about Amazon Web Services, or about all kinds of different technologies that relate to data engineering and they're indexed. You can find related links and comments and some community features.

Also, FindCollabs is the company I'm working on. FindCollabs is a place to find collaborators and build projects. If you have an idea for a project, you can post that project on FindCollabs and find other people to work with. We're also having a online hackathon with $2,500 in prizes. If you're looking for a project to work on, or you're looking for other collaborators for your project, check out FindCollabs and enter into our hackathon. I'd love to see you on FindCollabs. Let's get on with this episode.

[SPONSOR MESSAGE]

[0:03:10.7] JM: DigitalOcean is a simple, developer-friendly cloud platform. DigitalOcean is optimized to make managing and scaling applications easy, with an intuitive API, multiple storage options, integrated firewalls, load balancers and more.

With predictable pricing and flexible configurations and world-class customer support, you'll get access to all the infrastructure services you need to grow. DigitalOcean is simple. If you don't need the complexity of the complex cloud providers, try out DigitalOcean with their simple interface and their great customer support. Plus they've got 2,000-plus tutorials to help you stay up to date with the latest open source software and languages and frameworks.

You can get started on DigitalOcean for free at do.co/sedaily. One thing that makes DigitalOcean special is they're really interested in long-term developer productivity. I remember one particular example of this when I found a tutorial on DigitalOcean about how to get started on a different cloud provider. I thought that really stood for a sense of confidence and an attention to just getting developers off the ground faster. They've continued to do that with DigitalOcean today. All their services are easy to use and have simple interfaces.

Try it out at do.co/sedaily. That's do.co/sedaily and you will get started for free with some free credits. Thanks to DigitalOcean for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[0:05:11.4] JM: Danielle Morrill, welcome to Software Engineering Daily.

[0:05:13.5] DM: Thank you for having me.

[0:05:15.2] JM: You work at GitLab. Before we talk about MELTANO, which is what you work on specifically at GitLab, I'd like to talk about GitLab's initial product, which is called GitLab. We did a show about that a while ago and GitLab is an integrated monolithic stack for software development and has version control, logging, continuous integration, many other features, but the different features can be swapped out for other tools. There are lots of good tools that people can piece together to make their own ideal workflow, but oftentimes they will take GitLab for an out-of-the-box experience. Why do people use GitLab?

[0:05:56.4] DM: It's funny you're asking me this, because I operate very separately from the rest of the company. I will say I may not be the best spokesperson, but a lot of times it's either cost, or flexibility. Obviously, GitLab is open source. Sometimes people don't want to be using proprietary solutions. GitLab is really affordable option for a lot of teams that are just getting started with some of these features.

[0:06:16.7] JM: Yeah. I think what I was getting at was the model of a large integrated environment with the swappable components. It's not a common type of project in the open source world, because usually open source projects are like a narrow tool, a tool for configuration management, or package installation. GitLab is this large recomposing, which seems analogous to MELTANO. Why do you think this pattern of an out-of-the-box solution that is modular, why is that a useful pattern?

[0:06:50.4] DM: I see what you're saying in a connection to MELTANO. Yeah. I mean, in the case of MELTANO and in the case of GitLab, being able to connect things together is painful. It's not just necessarily snapped together in a lot of cases. Being able to offer an integrated end-to-end solution can become almost another product feature overarching the bigger picture.

With MELTANO, we are piecing together a lot of good pieces of open source software that you can totally use separately, but you may not want to have to administer, or may not have a person to administer the connections between that and similar with GitLab.

[0:07:23.3] JM: GitLab is for software development and deployment. MELTANO is for data engineering. How do those two domains compare to one another? Like deployment and release process, versus the process that a data engineer goes through?

[0:07:41.6] DM: Well, I think the deployment and release process is much more sophisticated in the classical software engineering world. I think data engineering, data analysis, that whole world crosses a lot more disciplines and it's still being sorted out, so what the workflow should look like different, depending on where you go. Data engineers often end up rebuilding pipelines from scratch. Each place they go, there's not as many accepted broadly used norms. We can bring some of those norms from software engineering to the data world. As we know, they save people a ton of time and a ton of pain.

[0:08:12.0] JM: How do most companies do data engineering and data science today?

[0:08:17.0] DM: Well, there's some big components. They have to have somewhere to store their data and that could be anything from a small server, to some massive tool. It really depends on the company. You fundamentally got storage, extraction of the data, you got to stick it in some tool to process and do analysis, you get the beautiful dashboard. A lot of people think it stops there, but actually a huge part of what's happening with data engineering is building some recurring process, or exception-based pipeline that's kicking off some other of business automation, or business process.

Right now, there is a huge proliferation of tools. There's not any one way. We think that we might be moving more in that direction of consolidation, you've probably seen some of the M&A that's happened the past few weeks with Tableau and with Looker, but there's a huge amount of tools in each one of those buckets. the market map for what's out there in terms of products you could adopt is massive.

[0:09:10.8] JM: What are the most acute pain points in data engineering?

[0:09:15.1] DM: I think a pain point that I notice the most with our customers is not so much of a technology problem, as a relationship problem. You've got a data engineer who's setting up a pipeline, but the end-user of the data is not the engineer himself or herself. It's usually an analyst, or an executive looking to make a decision.

The amount of steps you have to go through both from a technology perspective and from a communication perspective to get something that actually delivers value is pretty challenging right now. There's not a lot of reusable ways to convey business logic. I think you end up in situations where people spend a lot of time building something that then isn't very useful. There's just a lot of iteration and that whole process takes a lot longer than it should.

[0:09:57.9] JM: Remarkably, that was the same case four years ago, I think when I started the podcast. Why does it feel like data engineering, or data science is trapped – it's enmeshed in amber? I mean, it doesn't seem to evolve. We seem to have these same perennial problems. Has it been improving somehow?

[0:10:21.1] DM: I think it's improving. I think that it starts with the business, people becoming more comfortable trusting the data engineer. You've got to remember, we're dealing with probably the laggard part of a lot of these businesses in the case of the decisions this data is powering.

As much as I want them to adopt all this technology, I also can see why they might move slowly, because they may be doing things, like watering crops based on this data, or moving million dollar shipping containers, or moving cars. These are things where getting it wrong has these really real-world consequences. As a result, there's a lot of risk aversion. I think that's why it's moving slowly.

[0:10:58.3] JM: MELTANO is an acronym. It's a acronym for the workflow of the data lifecycle. What is the data life cycle?

[0:11:08.7] DM: Well first of all, I don't know, we might have to get away from this acronym at some point, so just a caveat that I think acronym names are interesting sometimes. The data lifecycle is just dealing with the idea of going from some raw set of data, or even sensors before the data is created, all the way through process of pulling that into some system where you can then analyze it. Human beings in the business process are at that point, probably looking at it manually and saying, “Okay, how do we model this? What does it all mean?”

Creating something repeatable and shareable like a notebook. Then ultimately, orchestration. The O of MELTANO deals with the entire automation process, which then is an entire other life cycle off in the business process world. It’s really, the data lifecycle is going from sensors, or raw data to some final output that can then be used to create value.

[0:11:56.1] JM: I think MELTANO is an acronym for Model, Extract, Load, Transform, Analyze, Notebook, Orchestrate. As we both know, the entire data engineering world is so gigantic that even these seven words that are acronymically arranged in to spell out MELTANO, this is not the full scope of data engineering.

My sense is that MELTANO is a tool that it's for a company that perhaps has several different databases. Maybe some of them are operational databases. Maybe some of them are databases where they've been throwing log data in. They've got a set of databases, maybe a data lake and they want to do things with that data lake, or those databases. Is that an accurate description of the day one customer?

[0:12:51.4] DM: Yes. Absolutely.

[0:12:52.9] JM: Okay. What are the prototypical users of MELTANO, like the people at this company that have a database and/or a data lake?

[0:13:03.1] DM: Yeah, so they're probably dealing with your information about how their business is operating, or how their customers are coming to them. The two buckets that we're seeing the most is business operations, where it might be we want to be able to basically build our own ERP. We want to see how everything that's happening in the company from product creation, to revenue collection ties together and we don't want to spend a fortune on something out of the box, or spend 10 months, I don't know how long the engagement is now, but consulting engagement to get that spun up. We want to do it ourselves. That's the business operations world.

A subset of that that we think we can really serve well is sales and marketing. The people who are saying we've got a lot of internet data about how people are engaging with maybe an e-commerce site, or a web funnel of some kind. Then we've got all those ancillary data we want to combine to give us an understanding of our customers and increase a goal, like conversion.

They can build those kinds of dashboards with some of the proprietary tools that are out there, but they're going to spend a lot and they're going to be really constrained in terms of the way these folks bill is either usage, so how much data are you processing, or seats. You have to limit how many people in your company can participate. Those are both limiting in different ways and a lot of companies say we're a huge business. We have a thousand people we want to be able to involve in this process and we don't really want to pay for a 1,000 seats. That's often why they turn to MELTANO.

[0:14:23.7] JM: The problem of sales and marketing needing better understanding of their data, are the sales and marketing teams equipped to solve this problem, or do they typically bring in a data scientist to help them with some back-end engineering, like standing up MELTANO?

[0:14:43.8] DM: Yeah. Generally, they will bring someone in. I wish I could say that I thought sales and marketing had folks you could do this themselves and I think that's the future. Today in big companies, you see they'll have a dedicated analyst who's very technical, or they'll have an engineer. You've probably seen auto companies to have the job title for economists in these teams. Those people can set up MELTANO.

MELTANO is not extremely technical to set up. You don't need to write a bunch of custom code, but it is nice and helpful to know how to deploy to a server, for example, because it is self-hosted. They will utilize that resource like that.

[0:15:16.4] JM: MELTANO, the acronym just to repeat it again, it's Model, Extract, Load, Transform, Analyze, Notebook and Orchestrate. These are not sequential, right? These are not in any particular order. These are just the different components of the data engineering lifecycle that MELTANO helps you with?

[0:15:36.1] DM: Yes. I think we thought at first they would be more sequential, but I think we've learned that they are not and that these happen in a lot of different orders depending on what's going on in our customer’s business.

[0:15:46.7] JM: The first stage of that data lifecycle is usually, I think to extract. You have the data sitting in your data lake, or your database. What happens in the extract stage?

[0:15:59.4] DM: The extract stage, well a lot of things can happen. Fundamentally, you want to get access to that data, but then the interesting thing is it can be in a lot of different formats. When you're extracting it, you're also preparing it to be put into a target, so in our case, a database or another format. Reading that data, being able to make sure it's clean, handling a lot of the exceptions that happen, but generally just getting the raw data out of whatever the source is is what's happening in that step.

[0:16:26.4] JM: After the data is extracted, it's often loaded into a tool, like a data warehouse. Can you describe the extraction and load process in more detail?

[0:16:38.2] DM: Let's see. What's going to be the best thing to explain here? I mean, I think the most important thing that's happening in the load process is that we’re then saying, okay, we need to understand what this is, so we're creating a schema and we're potentially also doing load and transform are very closely tied together. We need to be able to understand what we're looking at.

Generally, what's happening in MELTANO’s ELT process is you're choosing a tap or a source, you're choosing a target and they may not be the same format. You may need to do things in the middle. The load process is the preparation step. You might do some parsing. You might be reformatting certain fields, managing dates, managing strings, whatever, depending on what you target, you chose, but you're basically prepping the data and doing a certain amount of programmatic cleaning.

[0:17:20.1] JM: Why do people need a warehouse for data, a data warehouse? What purpose does the tool of a data warehouse serve?

[0:17:27.2] DM: I don't know about you, but I've definitely taken down my production database with complex queries before. The warehouse has a lot of values. One is just segmenting the data away from the production environment and maybe also, only taking a subset of that data. We talk about big data, and I think the truth is tools are very powerful now. We could probably pull down a lot of raw data, but we don't need to, and so performance is a really big piece.

The last is making it understandable. A lot of times in the warehouse, you will do things like defining what things mean, maybe you have a data dictionary. This is the place where we begin to say this is not data that's just meant for a machine to look at. This is now data meant for humans. The warehouse begins to make that understandable. In the past, people might have called the reporting database. Basically, it's just separating it away from the raw production environment.

[0:18:15.0] JM: Can you give an example, or two of operations that people might perform on a data warehouse and how it might serve a real-world purpose?

[0:18:24.8] DM: Sure. Let's just imagine that you're pulling in all your website data from Google Analytics and you're also pulling in all of your purchase data from Stripe. I'm just use a simple example. You've got some e-commerce site. The data warehouse is a good place to mash up that data in a very simple way, in terms of if you have unique identifiers, if you have duplicate data, if you have things in weird – I'm using simple examples, but you could have some much more complex formats. This would be the place where you would begin to rationalize how that data could work together.

It's still fairly raw. You're looking at it maybe for the first time. A lot of people don't necessarily run MySQL on their computer and look at the actual production database. This is often the first place you're seeing it. This is also where you might do things, like mapping, renaming tables, renaming columns, starting to make things make sense to a human being. I mean, that's a lot of the preparation process. Some of that is done programmatically, or can be done programmatically. Often, it's done by human beings who are trying to prepare that data to be consumed by other people down the chain.

The reason you might do this is I just listed two data sources, but you could easily have thousands and you need some sandbox to work with to do that. Generally, you're not merging these things together into production. You may not even be able to join these tables in a production environment. This is the first place where you can do that.

[0:19:44.0] JM: I think we've covered extract and load pretty well at this point. What about transform? Can you reiterate what happens during a transform? What is a transform?

[0:19:53.8] DM: Transform is the final step to getting the data into a usable format. You might say, "Hey, I've got this Stripe data. I've got this Google Analytics data, but I want to put this into a PostgreSQL database, or a MySQL database, or I want to put it into some proprietary third-party warehouse with some format requirements." That's the point we have to meet those requirements, often in order for it to be consumed by some other tool that's not inside MELTANO potentially. That's where you would need to actually do things to the schema, or do things to the data itself to make it conform to those standards.

Most of those things can be done without having to have human intervention. Fundamentally at the end, a successful transform is going to allow you to consume the data in the format that you intended.

[SPONSOR MESSAGE]

[0:20:44.9] JM: This episode of Software Engineering Daily is sponsored by Datadog. Datadog integrates seamlessly with container technologies, like Docker and Kubernetes, so you can monitor your entire container cluster in real-time.

See across all of your servers, containers, apps and services in one place with powerful visualizations, sophisticated alerting, distributed tracing and APM. Now, Datadog has application performance monitoring for Java. Start monitoring your microservices today with a free trial. As a bonus, Datadog will send you a free t-shirt. You can get both of those by going to softwareengineeringdaily.com/datadog. That's softwareengineeringdaily.com/datadog. Thank you, Datadog.

[INTERVIEW CONTINUED]

[0:21:40.1] JM: We can use data that's in the data warehouse to build dashboards and predictive models and do analysis. This would be the M and the A of MELTANO, the model and the analysis of the MELTANO acronym. What are people doing during the model and analyze phases?

[0:22:03.8] DM: Before we jump to that, I guess one thing I want to point out is MELTANO doesn't warehouse data. Since MELTANO is self-hosted, people are hosting their own data. I always found this very elegant. At first I was very skeptical, by the way, of MELTANO being self-hosted. It's very elegant, because MELTANO itself doesn't have to touch any PII, or touch your data, or really see your data. What's happening at the next step in the process is generally you say you want to have some insight. Let's use our Google Analytics + Stripe example again. Maybe we want to say, what is the relationship between visitors to your purchase page and revenue?

You want to get maybe average purchase value, or average value of a page view. Well, none of that data lives in just one place or the other, so you would probably build some nice little histogram saying, over the last few months, the average visitor to your purchase page, their value went from a $1 to $10, because your conversion rate went up. You want to see the relationship between data that was in these two different disparate sets.

The analyze step could be just outputting a bunch of results, but often it's also creating some visual. Most common use case for this in past lives was this is what I'm going to use to make a slide, or this is what I'm going to use to make a case for a change, or I'm going to send this table of data to some business person and be like, "Hey, I have support for something I want to do in the company."

The analyze step is really the step where we're trying to tell a story with the data and we're trying to do it in a way that we could actually go back and recreate the analysis if we had to, so that we can support our decision-making with facts, that's what's happening there.

[0:23:40.3] JM: What you described about MELTANO, not actually including a data warehouse, I mean, we've glossed around the edges of what exactly MELTANO is. I think it's a sequence of tools. It's a set of tools that help you frame your data engineering workflow more effectively. People might already have a database, a data lake, a data warehouse.

They might have some other things, but they aren't exactly sure of how to get the data from one place to another, how to eventually get it into dashboards. Could you just take a step back and give us the lay of the land for the things that MELTANO does not include, that it might sit over within a company that already has some software obviously?

[0:24:30.3] DM: Yeah, absolutely. It's a great question. The taps and the targets we've built, so the things that pull the data in and transform it are both from Singer, which is open source. For transforms, we're using DBT. If you're already using DBT, then you will be getting that bundled inside MELTANO, or you could use your own.

For notebook, we use Jupyter Notebooks. For orchestration, we're using Apache Airflow. The only two things MELTANO is offering that is MELTANO-unique right now are the way we're doing modeling of the data set. We have our own M50 file structure that you're using to describe the data model. Then our UI, which is a very, very basic, I would say fairly ugly, but functional dashboard.

Down the road, I think both of those, we'd like to replace them with the best-in-class open source. I think we haven't really decided what we think that is yet. For now, what we'll do is we're just making those as bare-bones as possible. Our goal is to not do anything proprietary that is going deep. We're much more focused on how we can be stitching things together at the thin flat platform layer.

If we can't find anything, I'd be shocked first of all. If we can, then of course, we could go and we could go deep in one of those areas. The reality is each one of these steps, there's multibillion-dollar product companies builds in each one. To think we're going to build a better dashboarding solution and someone who's spending all their time and resources on that is probably not realistic. That's how we divided things up today.

[0:25:59.4] JM: Coming back to the acronym. The N in MELTANO stands for Notebook. That is Jupyter Notebook. Describe what the purpose of a Jupyter Notebook is.

[0:26:13.4] DM: The Jupyter Notebooks are really powerful for doing analysis, if you understand how to write Python, or even if you just have very, very simple querying skills, there's lots of great copy-pasting you can do. Jupyter Notebooks are shareable, you can check them in and out with git, which is also really great. They're going to let you do much more sophisticated analysis than what our UI can provide.

If you are at all technical, you're going to immediately see MELTANO's analysis UI and say, "Okay, that's great for some pretty basic tables, but I need to do something much more complicated." Jupyter Notebooks makes a database connection to MELTANO. We provide it embedded as part of MELTANO, but you can also just use the database connection criteria we give you and go use it separately.

Also awesome, because you can run it locally under on your laptop and take it with you for analysis. You don't have to be connected to the internet. Very powerful tool and pretty much what most data analysts want to be using. If we didn't have this, I just don't think we would really be able to serve our core customer, because it's still the preferred way to do analysis.

[0:27:16.3] JM: Jupyter Notebook was not always the preferred way to do analysis. It seems to have really caught on in the last couple years. Why did it catch on and why did it take a while?

[0:27:29.2] DM: Oh, to me – we care so other people say, but I think it's this great prosumer product. It's powerful. It's really powerful, the same way that Excel can be very powerful, very simple depending on how far you want to go. The barrier to entry to using it has dropped a lot, I think in the past few years. I mean, I think people who never expected to write code and thought they would be analysts, but in a more Excel-driven world, are finding themselves using Jupyter very successfully. I think that's just a huge part of – it's also gotten better, it's got more connectors. They've been really fantastic with rolling out new features.

I think it's the right balance of useable and powerful and globally accessible, so it's not super proprietary. People are using it in all sorts of different environments. I just think they've struck a really beautiful balance.

[0:28:16.5] JM: The overall workflow of these different tools that we're discussing, things like the notebook, the Singer, DBT, all of this stuff is managed by an orchestrator, and you use Airflow as the orchestrator. Explain what the role of an orchestrator is.

[0:28:37.9] DM: Yes. By default, the orchestrator is not turned on. The simplest verbs use is just to think, “Okay, now I've got my pipeline working. I want to schedule it. I want it to run regularly every day, every hour, I want it to kickoff reports or business processes.”

The orchestrator is automating the process of stepping through the pipeline. It's also handling exceptions. You may have situations where one of your data sources produces corrupted data, or where there's not a target available, because maybe you haven't spun up an instance for your PostgreSQL DB, I don't know. There's million possible things that could go wrong.

We need to know what to do in the case that things don't go as planned. The orchestrator allows you to define all these different catches for what that could look like. A simple pipeline, you might be thinking, “Well, you just run it every day and get an exception report.” As people start to have hundreds of different data sources, kicking off dozens of different processes, you can imagine that that stack trace of what's going on begins to become really complicated. Apache Airflow is really built to handle that.

I would say most users right now, Airflow is maybe one of the harder things to adopt for them and I think there's some really interesting companies out there trying to abstract it a bit. We provide a really simple schedule verb, just to warm people up to the idea of orchestration, but for our most sophisticated customers, again, this is just something probably aren’t using, and so they expect it to work this way, or they would expect a tool like this to be involved. We just bundle all these things together, so you get them all out of the box.

[0:30:07.2] JM: When a company would – if a company was to adopt MELTANO and they've again, they've already got their database and/or data lake, they've got some data warehouse, maybe they're using Redshift or Snowflake, how do you – in an ideal world, how would their lives change after they started using MELTANO?

[0:30:30.9] DM: Well, I think big one is just they could try more things with their data. If they do have this data warehouse and they have a lot of things they've been wanting to explore from an analysis perspective, the adoption time would become much lower. Any target or tap we support would then be available to them. If maybe before they were limited by okay, we've got Redshift and Snowflake and that's all we know how to work with them, maybe they don't have a full-time engineer to build the connectors that they want. Whatever we have available is going to open those doors for them, which is a great argument for us building a really good community of people constantly creating more value.

The other piece is it's going to do some amount of the decision-making for them. I mean, you certainly can use whatever you want with MELTANO, but we do believe that we've chosen some of the best tools. As we grow and as we get bigger, we hope we can guide people down a path of least resistance to getting value, rather than spending a lot of time stitching this stuff together and choosing which tools and which steps.

In the beginning, when you really just need to get a business answer to someone, or you want to change, make an impact in some way, the time to value should be a lot shorter; that's the fundamental thing that should change, whether it's in the individual steps, or just in terms of kicking out something that's either an automated business process, or a dashboard at the end.

[0:31:44.5] JM: I think I'm starting to see the outline of the problems that you're solving here, because you think of a prototypical company where they've got this operational database. Again, maybe they've got a data lake where they're throwing their logs, but they've also got Google Analytics, which is sitting in Google. They've got Stripe data. Maybe that's how they're handling all their e-commerce purchases. That data is lying there waiting to be used, but it's not an accessible form. It hasn't been ported into the data warehouse.

You're solving that problem in a modular fashion, such that the numerous other analytics tools, or numerous other APIs and analytics tools and platforms that are gathering your data somewhat passively are now accessible.

[0:32:38.6] DM: Exactly. You don’t want to have to learn how to use each API, because there are so many. Each API is a data source. Google Analytics has an API, for example, but it's probably not actually worthwhile to become an expert on 50 different APIs. It's just not worth the time it would take for your team to do that. Instead you're just saying, “Okay, cool. This is people who have this stuff over here. You've adopted these SaaS tools.”

I think another piece by the way is with the proliferation of SaaS, you've also got a ton more places where data is living. Then you've got your engineers who have access to like you said, your logs and your more production data and a lot of them more nuts and bolts. Yeah, so it's just really division of labor and making it so MELTANO can carry some of that burden.

[0:33:19.4] JM: If I wanted to integrate my Google Analytics data, is there just a config file within MELTANO that I fill out?

[0:33:27.6] DM: Correct. Yes. You'd be giving us some of your connection information and then we would be connecting to the API on your behalf and pulling out your data.

[0:33:35.2] JM: Out of the MELTANO, again you have these defaults, these swappable defaults in each letter of the acronym. What have you had to build yourself and what parts are you taking off-the-shelf? Maybe we could just walk through the acronym at this point.

[0:33:53.4] DM: Sure. In fact, there's a great little table on the front page of our website that I'm looking at now to help myself talk you through this. You may want to look at as well, but yeah. In the MELTANO step for Model, we've built our own file-defining model standard. It's called an M50 file. We're actually exploring, by the way, kicked off after the Looker acquisition other possibilities here in terms of a more open model standard. Right now, this is something we're building ourselves in the M step.

For Extract and Load, we're using Singers, taps and targets. That's been working great for us, and so we just continuously are hoarding, or integrating what's already there that will come to some point where we’re either going to need to build a lot more taps and contribute them to Singer or start to build our own, since there's only so many.

For transform, we're using DBT. The Analyze step is MELTANO’s UI. As I mentioned, this is a very basic UI today. We're not really trying to be the next Tableau. We're not trying to necessarily do all those incredible visuals. We'd love to integrate really awesome open source alternative down the road. The Notebooks, using Jupyter. Orchestrate, we're using Apache Airflow.

[0:35:04.7] JM: What you mentioned about the – not necessarily I have the best – I mean, when I interviewed Cid about just GitLab itself, what's so interesting about this pattern that is being applied in MELTANO and also in GitLab, just the idea that let's just – let's build what we can, take what we can get off the shelf that's open source. It doesn't have to be perfect. Perfect is the enemy of the good. You just get something working, because that's so much better than cobbling together the things that maybe best-in-class, but are ultimately going to have a bunch of integration issues.

[0:35:41.6] DM: Yeah. I mean, I think you have to be humble about how hard these problems are. We want a working end-to-end solution. That's the most important thing every day. The question is are we making that end-to-end solution faster and easier for users? Each one of these steps is so interesting and could be the source of 10 years of work. There's huge businesses as I mentioned built in each of these spaces. It's tempting.

There's so many interesting technology problems, design problems, so it requires a little bit of discipline on the part of our team. We actually catch ourselves all the time going deeper than we should when we need to be going broad.

It's not that we shouldn't make things good, it's just you don't want to sacrifice the end-to-end value to make something pretty, or to add a feature that maybe a very small number of users are going to need. Or if they do need it, they should probably go with a proprietary solution.

[0:36:33.1] JM: Coming back to the acronym, with the M of Model, you use MELTANO model, right? For that phase?

[0:36:41.3] DM: Yes.

[0:36:42.4] JM: Could you refresh what happens, what is – describe what the Model step is and explain what MELTANO Model is.

[0:36:49.5] DM: The Model step is about creating reusable business logic. This is where you're defining the relationship between different pieces of data, or even just what they mean. This is after you've transformed your data and you have it in the format that you want, you start to need to put it into a table with headers, or a chart with it's a graph and it's got labels.

The headers and the labels while they are decorative, they're also really important because if you're looking at the data for the first time, or giving this off to somebody else, say you're the engineer and I'm the analyst. If you hand this off to me and it's not clear what's going on, I have to go parse the business logic. It's like what I was saying with learning each new API. Each one has a bunch of data in it. If you've used Stripe a whole bunch, you might be able to figure out what some of the fields are, but a lot of it is pretty esoteric.

The model is where you are turning this into something shareable and reusable. You're also defining possibly fundamental things, like what is revenue, or what is recurring revenue, or what is a contract value? You're also defining what the exception cases could be. That's really powerful as well, because you can control that. You can imagine checking that in and saying we're at the high level describing what the business means, what its dictionary looks like. Then everybody else who's also using that version of that instance of MELTANO has to conform to those models.

In big data teams, it's often a hierarchy, right? There's the senior people who are working with the data and defining at the core, what are we trying to achieve, whether it's a dashboard of KPIs, or going to produce an S1, or whatever that is. Then within those business rules, you've got many other analysts doing work. The modeling stuff is crucial for making sure we're all speaking the same language in the same company.

[0:38:32.3] JM: A model is a uniform object format that throughout different areas of the company we can agree on what is the shape of a certain object, what are the fields that it might have, so that different areas of the company can have a relatively consistent view of the data.

[0:38:51.7] DM: Exactly. You said it beautifully.

[0:38:53.8] JM: Okay. What about MELTANO model. Why was there not something off the shelf for doing – or what did people use off-the-shelf, or what are the alternatives to MELTANO model?

[0:39:04.6] DM: I don't think I have a great answer for you here. I mean, there definitely is – Looker has LookML, there's open model sphere. We list these on our site and also Matillion. I think this is one area. The places where we – MELTANO model and MELTANO UI were probably the places where I would say they are the most dispersed. There's so many different ways people are solving this problem today.

I mean, this is probably in my opinion one of the best reasons that Looker got acquired is because creating a way of thinking about this that businesspeople can understand is really powerful. I think that this is somewhere where we would like to continue the conversation now saying like, “Great, that's proprietary and that's awesome for the people who use Looker,” but what about just the broader world? I don't think there is a great solution.

We actually blogged a couple weeks ago on the MELTANO blog about a project that we've kicked off. I'm blanking on. There's a small company that had just tried to launch something very similar in the open source space. We'll make sure to drop their name, because we really were grateful that they reached out to us, but it's a company called Rakam, R-A-K-A-M.io. We started collaborating with them to define an open source alternative to LookML, which could be the replacement for MELTANO model at some point.

I think that what we're doing today is fine, but it's not necessarily going to be the best final solution for our customers. It's still a little limited in terms of what you can do in a file structure. There's may be other ways to approach it that could be more powerful. I'd say we're still very early in exploring what we want to do here.

[0:40:32.7] JM: There is a project called Singer that you use for, I think it's the extract stage?

[0:40:40.5] DM: Yes, and the transform stage as well.

[0:40:43.2] JM: The transform. I had not heard of Singer. What is Singer?

[0:40:47.0] DM: Singer it's an ETL company. They have both proprietary and open source solutions. We work with their open source taps and targets and they were actually – I want to say they were recently – were they acquired? They're sponsored by Stitch, so they're actually an open source project. Stitch is providing a much bigger offering now. I think they're acquired by – who is it? Talend, I think.

They're trying to begin to offer an end-to-end pipeline solution as well, but the Singer open source project is a subset of what they do. We contribute back to their community. I think Stitch probably uses that data as well, those contributions as well.

[0:41:29.1] JM: The Singer taps, what do they do in more detail?

[0:41:33.2] DM: These are the extractors that pull the data from the APIs and define what fields we're going to be pulling and how to format them.

[0:41:41.4] JM: Does the Singer ecosystem have these tools for if you want Google Analytics data, or you want Stripe data, is that something you can take off the shelf?

[0:41:52.7] DM: You could potentially take them off the shelf. The challenge is that they're not integrated into anything. You can take the tap, but then where are you putting the data? You can write it to a JSON formatted file, but then the question would just be where are you going to put that into? For the people who can write code, this is great. For the people who either don't want to maintain code or can't write it at all, then there's just this questionable, it's great, I've extracted this data, but now what?

[0:42:20.2] JM: Yeah. What is DBT?

[0:42:23.3] DM: Let's talk about DBT. We're using DBT for our transform. I guess, that's the simplest way to explain it. It's a way of defining what format you want the data. We talked about the ELT process. At the end, what we want is we want the data to be in a format that we can use. DBT is an open source community that builds a bunch of different tools for data transformation. As you can imagine, all this different raw data, you could have a lot of different things going on, you could have different data warehouses, you could have different exceptions.

We talked a little bit earlier about why would you have a warehouse? It's letting you skip that step, in the sense of creating a virtual warehouse where you hold all that data, you produce what would be the equivalent of a warehouse and you can use really whatever format you want and then you produce the transformation. Rather than staying in a warehouse state, you're actually getting the final result, the final transform data.

When you're using MELTANO, you're not necessarily spinning up a huge third-party warehouse in the load step, but you're virtually doing the same thing, and we use their technology to complete that.

[0:43:35.2] JM: You could use MELTANO without a data warehouse?

[0:43:37.8] DM: You could. Yes.

[0:43:38.8] JM: I see.

[0:43:39.4] DM: That’s why it’s so great, because it saves you a bunch of money. I don't necessarily know that you need one in a lot of cases.

[0:43:44.1] JM: Right. This is pretty interesting, because yeah, there's probably a lot of companies – I mean, what do you think – what's the threshold? Maybe if you have more than I don't know, some number of terabytes of data, or do you have any idea what the threshold might be where you start to need a data warehouse?

[0:43:58.1] DM: I don't know what the threshold would be, but I think the logic would probably be more about when you're doing a lot of things that are repeated and you just want to make sure – it probably is performance issue. At some point, it makes more sense to keep it than to spin it up every time. I don't know the answer to that question. I mean, I think it's much less expensive than it used to be, but it's another thing to maintain. Avoiding that, especially for a team that maybe is not the dedicated big data team, maybe they're just running biz ops process on the side, they probably can't necessarily get the $25,000 a year, or whatever they're going to need to keep that warehouse running.

I think it's very budget-driven in terms of who is the user, not necessarily how much data, if that makes sense. It's easy to get that budget if you're the normally the team that they expect to warehouse tons of data. I think it's a lot tougher when it's the FPNA team, you trying to run an experiment for three months and saying, “We want a warehouse terabytes of data. We want to spend all this money.” People are like, “Well, that's just not what you guys do. Why do you need that?” Does that make sense? It's more of a user-driven thing.

[0:45:04.0] JM: Absolutely. Yeah, it makes complete sense. Assuming we're talking about this prototypical case of a company that's adopting MELTANO, they've got some sales and marketing data that they want to integrate with their transactional data, there's a sales and marketing team, maybe a data engineer is coming over to help them set it up, what does that adoption process look like?

Let's say I work at some potential e-commerce company right now and I'm thinking about my data engineering problems and I'm thinking, “Well, this MELTANO thing sounds pretty appealing.” What am I going to have to do to get started?

[0:45:41.9] DM: You can get started with MELTANO pretty quickly in terms of just locally getting it up and running. We have a quick start with a Docker image. You can get going and have a really simple self-hosted instance in 10 minutes. I think the big thing that really drives adoption is figuring out what are the data sources and what are their business values. I think everyone has some data they can just hook up to, but the more interesting thing is what's the adoption process in the relationship between the data engineering and the business person they're probably partnered with?

Deploying MELTANO, the technology side is easy. Figuring out what you want to get at the end, that's the interesting and more challenging conversation. I think what often is happening is someone who can't parse the Google Analytics API or whatever, someone in marketing for example might go to engineering, or maybe they don't even go to engineering. They just go to their boss asking for resources saying, “Hey, we've got data locked up inside Marketo and Stripe and Google Analytics and 10 other things." They start messing around with Zapier, or other tools that they can use and pretty quickly, they just discover like, "Okay, this is really cool. We can do something really interesting."

We wanted to do it in a sustainable way that's using persistent data. I think that's really what actually is happening is there's this funny meeting point where MELTANO is not the first time you would have this problem and deploy something like this. It's probably just the first time you would do it in a way that then becomes a lasting tool, rather than a one-time fun experiment that was done by the marketing department with spreadsheet.

[SPONSOR MESSAGE]

[0:47:26.8] JM: As a software engineer, chances are you've crossed paths with MongoDB at some point, whether you're building an app for millions of users, or just figuring out a side business.

As the most popular non-relational database MongoDB is intuitive and incredibly easy for development teams to use. Now with MongoDB Atlas, you can take advantage of MongoDB's flexible document data model as a fully automated cloud service. MongoDB Atlas handles all of the costly database operations and administration tasks that you'd rather not spend time on, like security and high availability and data recovery and monitoring and elastic scaling.

Try MongoDB Atlas today for free, by going to mongodb.com/se to learn more. Go to mongodb.com/se and you can learn more about MongoDB Atlas, as well as support Software Engineering Daily by checking out the new MongoDB Atlas serverless solution for MongoDB. That's mongodb.com/se. Thank you to MongoDB for being a sponsor.

[INTERVIEW CONTINUED]

[0:48:49.4] JM: Eventually, of course this whatever prototypical company that adopts MELTANO to get their data engineering house in order, eventually they'll get sophisticated and they'll want to use things like, maybe Tensorflow, or some other machine learning tool. Is there any integration process that you've started to work on there, or is that a disjoint set of problems?

[0:49:15.7] DM: I definitely don't think it's disjoint, but we haven't worked on it yet. I think we have so much we need to do just in terms of making sure we get this pipeline piece working really well. You certainly can take the data that is a process through MELTANO and use it in those types of workflows or with different tools. Yeah. No, not yet.

Personally just given my own background with the machine learning team in my previous company, I would be shocked if we don't head that direction eventually, but there's so many problems worth solving in this data pipeline step that I think we've got our work cut out here, probably for the next year or two. If the community organically moves that direction and there's their contributions, they're certainly welcome, but we just haven't actually heard too much about that from our users yet. We're early, so we'll see.

[0:50:01.8] JM: Is it a best practice to typically have data go through a data warehouse before it lands in Tensorflow? Because I could imagine you want to do a bunch of pre-processing or something before you put in a machine learning framework, or is there just not really best practices around that?

[0:50:17.5] DM: Well, I think there certainly are. I mean, I think if you're talking about cleaning a training data set, then yes, absolutely you would probably want to produce a clean set, just because you're trying to manage a lot of different things at once. It's nice to not have additional exceptions coming from dirty data. Yeah, I definitely could see building a data pipeline that's just doing cleaning and removing of things that are going to create issues in your experiment with MELTANO. It's definitely something you could do.

We just don't have the connector at the end. I guess, what I what you could say is you could use Airflow to kick that off. You could have orchestration, send it over to Tensorflow. We just don't really think of that as part of – Orchestration is like saying, "Okay, anything you do after this step, it's cool. You can hook up anything you want to Airflow, but was out of our hands, I guess at that point, at that hand off point."

[0:51:07.8] JM: Yeah. I mean, you could probably say the same for – I mean, I think the data warehouse is like Snowflake, probably has a lot of integrations to something like Tensorflow.

Let's talk about the management in the software development process. You're the general manager of MELTANO, what does that job entail?

[0:51:25.2] DM: Great question. It's changing all the time. Right now, I am the engineering manager, program manager, the product manager, the marketing manager, the sales manager. I have four engineers on the team. They're all very senior and wonderfully self-managed, I would say, I'm very grateful to them. Yeah, right now right it really it just comes down to prioritizing what we need to do next. I spend a ton of time dogfooding, writing issues and talking to the community, blogging, making sure that what we're getting, what we're building gets out to the world.

I'm definitely still pretty down in the details with the team. With four people, there's only so much we can do each week. Choosing. We do a release every Monday, so choosing very carefully what we do so that there's incremental value every week is really the name of the game. I've been on the team – about four months before that, this team was led by GitLab CEO, who I think you spoke to recently. My role is really to remove that distraction from him and keep the project moving forward and remove barriers for the engineering team.

[0:52:25.2] JM: How was this project initially ideated? Who had the idea to just say, "Let's apply the same lessons of GitLab to data engineering."

[0:52:36.4] DM: Well, my understanding is it really began as an internal project at GitLab. Over a year ago, as the company began to scale they had of course all their own sets of data and dashboards they wanted to produce. I think it started out as something where it was like, "Well, let's solve this problem for ourselves." As we looked around at different tools that could have been purchased realizing like, "Oh, man. This is actually going to be very expensive to solve with other software."

Now to be fair, it turned out MELTANO was so early and GitLab is growing so fast that GitLab actually needs to use other tools right now, because we have a board to satisfy and we have a lot of internal stakeholders. I think that really indicates where MELTANO is at in our development. That is the genesis of the project. MELTANO today is probably best adopted by slightly smaller companies that are extremely scrappy, trying to save money. GitLab is in a place where it's just growing super-fast and needs a lot more than what we've currently built.

[0:53:32.6] JM: Describe the division of labor among your team.

[0:53:35.3] DM: Well, I try to basically make it so the engineers can just write software all day, if at all possible. We have very few recurring meetings. We've got a rough road map, but truly we're trying not to lock in too far into the future. I would say I'm a janitor, in terms of I manage the milestone planning and the main meetings and spend a lot of time writing very detailed bugs and detailed dogfooding issues, so that we can fix things, add polish.

We have two engineers who focus on the frontend. We've got the engineering team layered in terms of layers of abstraction of the product. We've got an engineer who's outward-facing, marketing-facing, managing the site, the docs and a lot of the polish for the UI. We've got a friend of an engineer who truly, he's more full-stack, but I would say he's serving as our designer and he's building most of the MELTANO UI.

Then we have two backend engineers; one who's more on the data engineering side and one who's really just a full stack backend engineer. That's a lot of wiring up how we serve what originally was a command-line product. We build everything at the command line level first, serving a layer, an edge to the UI, so that we can integrate all those features. Yeah, it's a full plate for each person involved.

Then we also have, I'd say two or three more regular community contributors right now who are I would say making MRs every couple weeks, and then a few other dabblers. I'd say at any given time, the team is maybe four to six engineers in a given week. They're making merge requests against the project.

[0:55:11.3] JM: What's the biggest challenge you've encountered so far?

[0:55:14.0] DM: I mean, I think really at this point I just want to go faster. I think it's still what I was saying at the very beginning of our call, the discipline to continue to go broad when it feels sometimes you want to go and polish and go deep. I think that is not intuitive. I think it's much more satisfying to go and just craft something and hone it and make it perfect.

We are walking through each step of the MELTANO acronym as a team, touching each step, making it better and moving on and trying to make that rotation at N plus one cycle as fast as possible and it's painful. Because sometimes you just get to a step where you're like, “Oh, but this is still ugly. I don't want to move on. Or this isn't useful yet, I don't want to move on.” We're finding that we ship fastest if we continue to make that progression happen over and over again.

[0:56:01.7] JM: Much of what we have discussed is about engineering and tooling. As you mentioned earlier, the most acute problem might be communication. How should data engineers be interacting with software developers and other members of the world, like sales and marketing teams?

[0:56:26.2] DM: Yeah, that's a tall order. I think this is probably going to be where the core of our marketing resides. What I’ll say is you got to talk in stories, because it's got to have a payoff. A pretty dashboard with no point, just doesn't really matter at the end of the day. I think that's probably the place where people struggle. There's things that are technically really cool, but they don't add a lot of value. There's also things that are curiosities in business, but don't really drive end results. Getting good at talking in stories so the end is in mind before you embark on building these data pipelines is I think, someone everyone in that relationship is responsible for doing better.

[0:57:03.9] JM: What's something that you've learned about the world of data engineering that you didn't know before you started working on MELTANO?

[0:57:10.9] DM: I think, I thought expected certain things to be more standardized than they are. I mean, that we talked a bit about the model files. I also think the world of APIs is far less standard than I expected, the way we talk about and think about different pieces of data. I think I had lived a little bit in my previous role, I'm a CEO. I was a recipient of a beautiful analysis and beautiful dashboards. Even though I was able to go and inquiry the data myself, I just didn't really understand, I don't know, I think the idea of big data is hard to hold in your head, just what it means to – what big means. It's not just the quantity, it's also just the diversity.

As we work out all these different taps in particular, it's like building printer drivers; every single one is a little bit different. I think that adds a lot of complexity and I think it is to me, screams opportunity; opportunity to solve annoying problems that most people don't want to think about day-to-day. I just didn't really realize how fast that was.

[0:58:09.2] JM: Danielle, thanks for coming on Software Engineering Daily. It's been really fun talking to you.

[0:58:12.4] DM: Thank you.

[END OF INTERVIEW]

[0:58:17.1] JM: Commercial open source software businesses build their business model around an open source software project. Software businesses built around open source software operate differently than those built around proprietary software.

The Open Core Summit is a conference for commercial open source software. If you are building a business around open source software, check out the Open Core Summit, September 19th and 20th at The Palace of Fine Arts in San Francisco. Go to opencoresummit.com to register.

At Open Core Summit, we'll discuss the engineering, business strategy and investment landscape of commercial open source software businesses. Speakers will include people from HashiCorp, GitLab, Confluent, MongoDB and Docker. I will be emceeing the event and I'm hoping to do some onstage podcast-styled dialogues.

I am excited about the Open Core Summit, because open source software is the future. Most businesses don't gain that much by having their software be proprietary. As it becomes easier to build secure software, there will be even fewer reasons not to open source your code.

I love commercial open source businesses, because there are so many interesting technical problems. You've got governance issues. You got a strange business model. I'm looking forward to exploring these curiosities at the Open Core Summit and I hope to see you there. If you want to attend, check out opencoresummit.com. The conference is September 19th and 20th in San Francisco.

Open source is changing the world of software and it's changing the world that we live in. Check out the Open Core Summit by going to opencoresummit.com.

[END]