EPISODE 1172

[INTRODUCTION]

[00:00:00] JM: The JavaScript ecosystem has millions of packages. How do you choose from those packages to find the best of breed for your projects? Openbase is a system for searching and discovering JavaScript packages. Openbase includes reviews, insights and statistics around these JavaScript packages. Lior Grossman is a founder of Openbase and joins the show to talk about the JavaScript ecosystem and what he's building.

If you want to support Software Engineering Daily in a greater capacity, go to softwaredaily.com and become a paid subscriber. You can get all of our episodes without advertisements.

[INTERVIEW]

[00:00:38] JM: Lior, welcome to the show.

[00:00:39] LG: Thanks for having me.

[00:00:41] JM: Simple question, why are there so many JavaScript packages?

[00:00:45] LG: Well, I think it's about like developers having the possibility to create and publish something so easily in order to help themselves and other developers. I think the barrier to releasing and publishing packages, getting them to thousands of other developers has like become quite easy. And that means that over time more and more developers kind of started encapsulating some of the problems that they've been solving for themselves and helping other developers by packing these packages. And so like publishing them on NPM, allowing other developers to basically enjoy these same things that they built for themselves instead of like reinventing the wheel.

[00:01:33] JM: Do you have any information about the rate at which JavaScript packages are growing in number?

[00:01:39] LG: So I don't have any particular rate, but I would say that like we just saw like I think last year NPM I think mentioned that JavaScript packages has crossed like 1 million. We're already at like almost 1.4 million. So that rate is definitely growing exponentially. I would say that's kind of like unprevalent. We do expect to like see the number of packages in NPM and maybe in other ecosystems as well exponentially growing over time.

[00:02:09] JM: And what kinds of issues come with the fact that there are so many JavaScript packages?

[00:02:14] LG: So I think the biggest problem is that like having choice is a good thing essentially, but developers I think over time they grow tired of like having to make these choices. Like every time you have to select a package you have to exert some amount of energy, you have to spend a lot of time researching the different options, different possibilities, evaluating the different packages. See if they fit your needs. And over time this causes what they call nowadays like JavaScript fatigue, as in developers become exhausted of like having to choose between so many options. So in a way, having too many options isn't necessarily a good thing.

[00:02:58] JM: Are there any safety concerns with the fact that there are so many JavaScript packages?

[00:03:03] LG: I think there are like many concerns. Safety I would say is one of them, because when you have – For example, like a typical modern web app uses probably dozens of different packages. And one of the problems with that is that every time you embed an external package into your project you basically run the risk of like triggering like licensing issues, basically embedding something that causes like a security vulnerability within your project. And I think the more people use open source packages, the more they sort of like become susceptible to these licensing and security issues.

Setting that aside, there are like many other potential issues such as like installing a package. Starting to use that package and then like it becomes unmaintained over time. And then like even if you had some kind of security vulnerability, you have no one to maintain that package, no one to alert you, no one to fix and remediate this problem. So I think this becomes a real issue over time as we have like so many package, many of them are essentially unmaintained over time.

[00:04:13] JM: You're working on Openbase. Explain what Openbase is.

[00:04:16] LG: So Openbase is a website that helps developers choose among millions of open source packages. We found that developers waste hours researching and evaluating competing packages and we want to help them choose the right package every time, and we do that with user reviews, and we do this with powerful insights that they can't find anywhere else.

[00:04:40] JM: And give a little bit more detail on what Openbase does though. If you go to the website, what does it offer you?

[00:04:47] LG: So if you go to Openbase, you can basically do a couple of things. First is you can search for open source packages by a name, for example, and then you would go to a package page for React, for example. In the package page for React you would get deep insights about like this package, but you can do the same about like a million other packages. So you could get things like average time between versions, minor versions, major versions. You could see how long on average it takes to merge PRS and issues. You could see the star count over time. So all these metrics come to help you choose the right package essentially.

And within this package page you could also read reviews by other developers who've used React. So that is one way of like browsing Openbase, namely by using search. The other way which is kind of like more unique is that we've actually done the work of categorizing a lot of open source packages. That means that you can browse our catalog of different categories.

And for every task that you want to perform, for example, you're looking for an NLP library for Node.js. You can go to Openbase. Open the category browser, search for that specific category and basically find and compare the different NLP libraries for node. And we have thousands of different categories to choose from.

[00:06:09] JM: What is the incentive for people to write reviews for these packages?

[00:06:14] LG: So I think it kind of happens organically in the real world where often times when a developer needs to pick a package, they often times they chat with a friend, they pick up the phone, they ask on Reddit or Facebook or they post something on social media. So I think developers kind of want to share their experience like working with different technologies with other developers. And developers oftentimes look for the experience of other developers, basically learning from the collective knowledge of like other people before they choose a package. So I think the motivation is largely intrinsic in a sense that like developers, they have strong opinions about like different pieces of technologies that they've used. They were like disappointed or they really like enjoyed using it. Maybe the documentation was great. Maybe it was lousy. Maybe they found the package to be like super performant. Maybe it was slow. They usually want to share these experiences with other people. So I think it's largely intrinsic.

And I think there's another thing at play here, and that is like developers sort of like love expressing themselves in terms of like building different like pieces of software, but I think in sort of a way there's like a pride in using specific technologies. And I think developers overall like they take a lot of pride in choosing and using the technologies that they love. I think they want to share it with other people as well.

[00:07:45] JM: Has Openbase, building it and becoming a user of it, has it revealed any insights to you about the JavaScript ecosystem that you wouldn't have known otherwise?

[00:07:55] LG: Well, we did have a few observations building Openbase obviously. I think one thing is about star count, which I think you know a lot of people use GitHub stars is also like way of measuring projects like popularity. And one of the first things that we build in Openbase is we built this graph that shows the star count over time. And then we figured like what's wrong with using star count? What is wrong is actually no one is ever unstaring a package. That really rarely happens. And what happens over time is the stars continue to accumulate and no one is actually unstaring a package. So you could see a package, for example, that hasn't been very popular even for like the past three years and yet you would say it has like many stars over time.

So one thing we found out is like that metrics can be misleading. And one of the problems that we're still struggling with and trying to solve is finding the right metrics to help developers choose the right package. Another example is the download count. So you would think that downloads are great. Like if people don't use a package anymore, then the download account would go down. And if more people would use a package, then the download count would go up. But then again looking at the numbers what we understood is that it's not a good metric, because oftentimes developers download a package because it's a sub-dependency, it's an intrinsic dependency. And they don't even know they use this package. But some other package that they've been using actually install this like sub-dependency. And then like looking at the download count you would think, “Wow! This package is amazing. It's like super popular. So many people are using it.” When in fact very few people even know about this package, because it's only installed indirectly.

So I think our biggest take from building Openbase that you cannot look at metrics at par value necessarily. Like they don't tell the full story. You have to dig deeper inside in order to find the kind of like deeper insights as to like how well-maintained these package is and how popular this package really is essentially.

[00:10:16] JM: It's 2020. How is JavaScript development changing? How is the process of developing a full stack JavaScript application changing?

[00:10:24] LG: I think, paradoxically, JavaScript development should have become easier over time as we have more tooling, more technologies, more scaffolding to help us build apps. But I would say counter intuitively, JavaScript development has actually become quite harder. And I think one of the reasons is that it's become so complex to build an app. Or like back in the days you could just like shoot up a text editor and write about a bunch of like HTLM and JavaScript and just run the app in the browser. And nowadays like you have tooling place that you have to do. Like you have to use something like WebPack or parcel. You have to set up like testing. You have to set up CICD. And basically everything has become more complex, and we're kind of like seeing this trend evolving. The more tooling there is, the more complex it is to develop a JavaScript app. And I think that is like unfortunate in a way, because that's all like puts the barrier higher for like junior developers to start working with JavaScript. They have a lot to learn. Where like back in the days, they could start learning JavaScript and just with some code examples. So like shoot it up within like an hour. Nowadays there's a lot of boilerplate around it.

On the upside I would say we do see a lot of tools helping with that, basically, boilerplates, different like website builders, all sorts of framework that so like help you ramp up and launch your product to market in a matter of hours as opposed to like maybe days or weeks.

[00:12:03] JM: Tell me a little bit about building Openbase itself, the website. Is there anything that has been particularly difficult in building the site?

[00:12:10] LG: So I think one of the hardest things for us was dealing with such an immense amount of information, because essentially in order to build Openbase, we have package pages and category pages for 1.3, 1.4 million JavaScript packages. And for each and every package we essentially have to read through the entire commit log, which could be huge. We have to sift through all the issues. We have to see through all the PRs, and namely reading this insane amount of information and storing it and making sense of it and basically trying to extrapolate interesting metrics out of it. I think that this was like a big challenge, A, I would say in terms of like making it reliable. I think one challenge is making sure we have an information that is up-to-date that every time a developer visits Openbase, they get like the most up-to-date like picture of this package.

And the other thing is about like querying all this data, like we're trying to extract like pretty complicated metrics from all the commits and issues and PRs and like the list of maintainers and different people. So a lot of the heavy lifting was around that. I would say the second part is actually the manifestation of that. Like even once we have the metrics. So we have this time series, building the right user interface to convey that information. One thing we're still struggling with is information overload, namely there's like so much data that we want to display. So much data that could be helpful for like developers. And I think this poses a challenge in terms of like performance, in terms of like the user experience, the information architecture. So it's not enough of a challenge to sort of like get all this data. It's quite a product and user experience challenge actually displaying the right pieces of data in the right time to the right user.

[00:14:12] JM: How often do you crawl the repositories of data that you're crawling like GitHub?

[00:14:17] LG: So basically we crawl them all the time. And in terms of like our SLA, we strive to have all the data up to date within a 12 hours period. That means like he would never go to Openbase and see a specific metric or something that is like more than 12 hours old. And that is quite a challenge in terms of like we do have to run a lot of servers, quite a huge database and do a lot of crunching in order to like get all the data. But it runs all the time 24 and 7 repeatedly.

[00:14:51] JM: Are there any batch jobs you have running across the site?

[00:14:55] LG: You mean like cron jobs, I would assume?

[00:14:57] JM: Yes.

[00:14:58] LG: Right. So we do have some jobs running on the site. For example, you could log in to Openbase, for example, when you write a review or you want to save a package, for example, add it to like your own favorite packages. So we have one job that runs on the website in order to like gather the activity done by users and send all kind of user's notifications. So one example of that is like you could install and NPM package on Openbase and over time we're going to send you this notification about like how about you rate this package?

One other thing we're working on right now is basically providing a better like digest for users, and that essentially means that we learn over time about like your favorite stack, your favorite packages, and we are building this job in order to send you a concise summary of things you might be interested in. For example, new major versions of packages that like you've previously installed, or like other packages that are kind of similar that you might want to hear about. So that's one class of jobs that we have.

And the other thing is obviously we have a lot of workers on the backend basically working to obtain all the data that we need from GitHub, from NPM, from other sources and crunching the data. And basically we're kind of dependent on these jobs to be able to provide fresh data to our users every time they visit Openbase.

[00:16:30] JM: Tell me about the direction of the business. Where do you see Openbase going and how do you expect to monetize it?

[00:16:37] LG: So in terms of the direction of the business, like as you could see, you go to Openbase right now and we don't monetize it at the moment. What we are seeing though is that like right now we have almost 400,000 developers a month visiting Openbase and the intent is very clear. Like most of them come from Google. Most of them are looking for like specific pieces of technology. They try to solve a problem. And we help them find the right package. So we think in a way we're at the decision point where developers make these decisions about technologies. And we think over time – We're not monetizing right now, but over time we do think a lot of companies, for example, API companies, SaaS companies, databases, monitoring tools, basically all these companies, they would want to get the mind share of developers. And I think in terms of intent, as developers visit Openbase every time they're looking for a new package or a new piece of technologies, they're trying to choose like a database, search solution for the website, for example. I think these companies would love to get the mindshare of developers. And we do see ourselves like over time allowing these companies to promote their own products on Openbase. Obviously, we're not going to touch any of the integrity of the data. Like the metrics you get are the same metrics. The user reviews are the same user reviews. But we think letting an incomer, like for example an open core open source company just want to go to market. They don't have any other way of like promoting their product right now. They cannot go and promote their products like actively on GitHub or on NPM. So usually what ends up happening is they spend a bunch of money on like content marketing, which is great, but I think Openbase can be even a better outlet to help this like open source, open core companies, API, SaaS companies, get the mind share of even more developers.

[00:18:39] JM: How have you seen the Vue communities and the react community diverge in terms of who goes there and the usage of those two different frameworks?

[00:18:51] LG: So I wouldn't say we have a good view in terms of like the segmentation of the market and when it comes to like Vue versus React. One funny thing I would say is that like in terms of like looking at user behavior, you could say that it's very polarizing. For example, we've seen like many people who like went and rated like React. They were like raving about the framework. They would usually end up like writing a pretty bad review for Vue and vice versa. So like in a way I would say the market is very polarized. I think more than like even thinking about like feature set of philosophy, I think it does become so like a religious issue. The same as like you had like theme versus Emacs, or like tabs versus spaces. So I think everyone has a pretty strong opinion about what they like. But I wouldn't say we have seen like any very distinct segmentation whereas like one type of developers choose React and other type of like developers choose Vue.

[00:19:59] JM: Are there any ways in which people use Openbase that has surprised you?

[00:20:03] LG: Well, one thing we've seen early on, which was like it was so like surprising to me is that every time we've seen like a package maintainer. So like using Openbase, they ended up like inviting a bunch of other friends to so like look at Openbase. So one thing that was surprising to us that we didn't realize early on is that like maintainers are so like evangelists for Openbase, and every time they see their package page in a whole new way, I mean it's quite different that like looking at the NPM or GitHub page or like some other pages. They would so like promote it. They would so like tell other people's about Openbase. A lot of them has reached out to us. A lot of product feedback as to how we can improve like the maintainer experience.

So I think this was kind of surprising, because essentially we built Openbase for the developers, for the users first, but we did find out that like maintainers use us quite extensively. A lot of package pages have been already claimed and we've actually – Based on that feedback, we built some tools for like maintainers to be able to like put their packages in the proper categories. And we've even introduced like a badge for like package maintainers. That has been, I would say, quite surprising, because for us this was not the core of the product. The core of the product was like the individual user that wants to perform this like one specific task and they go to Openbase. We help them choose the right package. Whereas we found the other side of the marketplace is just as engaged and active as like the package users.

[00:21:52] JM: So it seems like you're kind of able to turn this into something like a social network for packages.

[00:21:59] LG: I guess you could say that. And I think in a way when people look at Openbase, they still look at it more as a utility I would say, more as a utility than a social network. But we definitely have some social elements in place and we found that users really love these. For example, we recently added the capability to up-vote and down-vote reviews. And we're working on some stuff that would allow users to like even contribute like more content to the community and discuss around that content. We've allowed people to so like upload resources such as like tutorials and videos and other stuff for these packages.

So I think you could think about it like as a social network. But the way I see it, it's more about like we have I think in the JavaScript ecosystem about like 1.3 million packages. So I think we sort of want Openbase to become this center where people go every time they're like interested in this package. Whether we're talking about choosing a package, but also about like learning how to use this package. You have like tutorials or like seeing code examples of this package. Looking at videos or like screen search of this package. So I think becoming the epicenter for each and every one of these packages, that's how we kind of see Openbase and a lot of social elements to it, definitely. But I would say it is still a utility. Like it's a work tool basically for developers that helps them become more productive, choose the right packages and use them effectively.

[00:23:40] JM: Have you considered making Openbase itself open source?

[00:23:43] LG: So yeah, that is actually one decision that we had to make like early on when we just started Openbase. I would say that was kind of a tough one. Eventually we decided not to go with it and not to open source Openbase just yet. The reason was that like we expected just like every early stage startup, do we have a lot of pivots? The things would kind of like change over time. We kind of assumed over the first couple of years we're going to have like many direction changes, product changes, user experience changes, which I think is very confusing where you like publish an open source project where you need to have certain directions, certain roadmap. We didn't want this to be an inhibition. We didn't want this to have any like negative effect on the community. But as like Openbase matures and stabilizers, we do see ourselves like over time opening up an API, opening up some parts of like the codebase open source as it becomes more stable and mature.

[00:24:49] JM: What do you think about the low-code community relative to the Jamstack community? There's kind of the promise that both of these can afford much easier frontend development, but they kind of come at it from different angles. Do you see a merger coming between low-code and the Jamstack?

[00:25:09] LG: So I think they're like kind of running in orthogonal directions. And obviously sometimes like there's an opportunity cost. Like as a company, sometimes you have to make a decision and you either like decide to go by building something with Gatsby, for example, or like you go on and build something with a low-code solution. So I think we do see some merging in the space. For example, you see companies like Stackbit kind of produces something that is like the best of both worlds, namely like you can still use the Jamstack technologies. You can use the CMS. You can use like stuff like Gatsby, but still you make it friendly enough so that the people who use it, they can be developers, but they could be marketers as well, I would say.

So I think it's not about having this inevitable merge. I think some people would opt to go like all the way building a Jamstack website. Whereas like for other companies, other things, it would make more sense going for like simple to use maybe like low-code solutions. But I do expect over the next few years that we see more like companies, more solutions. So like integrating both approaches, which i think when it comes to no-code, I don’t see it as like a binary thing. I think it's about like library to entry. And I think even when developing Jamstack solutions, you could see the entire spectrum from like having to write everything yourselves from the ground up through stuff like scaffolding tools and boilerplate tools through like website builders and other utilities that basically enable everyone to build their Jamstack website.

[00:26:58] JM: So in terms of indexing open source projects and making reviews for them available, are you only focused on the JavaScript ecosystem or do you plan to go to the rest of the market?

[00:27:09] LG: So we actually started working on expanding to Python and Go, and we started with JavaScript because like I started Openbase basically out of my own pain as a developer having to like research and evaluating competing packages. And I think the pain is most notable in the JavaScript ecosystem that has so like exploded exponentially. But I think you do have the same pain with other languages. You have the same thing in Python. You have so many like different projects in Go.

So we want to use the same methods that we use on the JavaScript ecosystem, namely provide deeper insights to help you choose, provide user reviews and try to categorize like this entire like registry of like so many packages. And yeah, we're about to expand to Python next and then Go, but essentially we want to cover like as many different languages as we can, because we think this problem – Well, while we started with JavaScript, we think it's not exclusive to JavaScript at all. We think it's kind of like decision fatigue plagues even other ecosystems as well.

[00:28:20] JM: So it'd be great for your business if you could become the de facto place for JavaScript package reviews, maybe even other open source project reviews. And in order to become the source of truth you obviously have to be the source of truth according to Google. So what's your interaction with search engine optimization?

[00:28:40] LG: So, funny thing is we launched Openbase just over a year ago and we grew quite rapidly. I would say primarily thanks to Google. But I think the funny part about that is we never optimized for SEO directly. We didn't use any kind of like SEO tactics to get some traffic. I think what we did that was like super beneficial was kind of talk to a lot of developers and understand what they're looking for instead of like in terms of like every time they look for like a specific task they want to perform and they look for the right package, what they're searching for? What kind of information they need?

So we kind of optimize for that. We optimize for the user intent and we optimize for the user language by speaking to many developers. And we've interviewed like dozens of developers. We find a kind of like language and terms that like developers usually use. And then obviously we had to validate that. So we use different tools. We use like Google's keywords research tool to so like validate that this is indeed what developers are looking for.

And I think by simply providing this content that people are looking for in a market where like you couldn't find this content, I think this was picked up by Google and we started getting nice amounts of like traffic over time.

[00:30:06] JM: What kinds of terms do you find people searching for the most in Openbase?

[00:30:11] LG: That is quite interesting. So we actually see mostly like two classes of searches. First one I would say is probably name search where like people look for a specific package. Where, for example, you just installed this package and you're so like breaking your head like you're having many troubles getting it to work or you don't really understand how to use it. So people could look for like something like React tooltip and then like troubleshooting, or React Tooltip alternatives, or React Tooltip name of like a specific problem. So in all these cases a lot of people search Google for like over a million packages. A lot of people use these name searches. And basically when they land on Openbase, we send them to the package page. And in the package page they can get basically the latest documentation. They can talk to other developers. So we kind of help them resolve their problem, but we're also showing them alternatives, which means for this specific package you might want to consider those other like 10 packages that could be good alternatives. And then obviously we try to help them make it all informed choice.

So this I would say is the first class of like search that we were seeing from Google. The other type of term is people looking to solve a specific problem or whether like they're trying to tackle a specific task. So, for example, someone could search for like React auto suggest. So what you usually see in this case is people looking for the task's name, but oftentimes they would add something like a specific framework. So it could be React auto suggest, but it could be like Angular or Vue auto suggest, or like node search engine or something like that. And in this case they usually land on our category pages.

So as I mentioned we have tens of thousands of like category pages on Openbase. We do a lot of work kind of categorizing these packages, because we want to provide the right experience. I mean when a user makes this kind of search, they are basically looking for something like the best solution to help them like resolve this package. Now typically they would go on NPM and read the readme and so like make some other search or they would go on GitHub. They would so like go through the readme and then go over like the issues to see if like the package is well-maintained. Go over the commit log to see what's that development velocity like and then eventually decide about the package.

People will come with this intent. They want to reach a decision and they want to reach it like as soon as possible. So we help them by trying to like put all the metrics in that one place. They would go on this package and we show them like 10 or 15 different like auto complete packages and you can compare like the different things, like the maintenance only package, the popularity of each package over time. And then essentially we want them to be able to make a decision to reach a decision and install a package as soon as possible. I would say these are the two main classes. So like name search or people looking for a specific task that they want to perform.

[00:33:34] JM: So in some ways it can be like a Stack Overflow.

[00:33:37] LG: In a way it can. And although we don't have any specific question and answer capabilities on our website. So I think in such case if people have like a very specific kind of question, oftentimes they can get the answer on like Stack Overflow. Whereas in other cases where they have like an open-ended problem or they're just like frustrated with some aspect of this package. For example, it's too slow, or like it has a vague or immature kind of like documentation. Oftentimes they end up on Openbase and we help them you know assess the other alternatives for this package.

[00:34:19] JM: Have you had to do any community management like to reduce spam or anything?

[00:34:24] LG: So we actually actively monitor right now all the reviews on Openbase and all the tutorials to make sure they're all legit, there isn't any kind of spam, there's no like solicitor reviews. Because I think in a way we were like the first to build like this kind of like reviews platform for open source. And I think in a way it's a lot of fun, but it's a huge responsibility, because we cannot steer the conversation. We cannot have like spam on our platform and we got to make sure that like every review that you read is kind of trusted.

So we actually go over like each and every review that goes up to Openbase before it's online. I would guess as we scale, as we become like a hundred times larger, that would become quite a big issue in terms of like moderating this content. And I think at this stage we would probably appoint people, basically ambassadors from the community, to help us moderate this content. But right now we're doing everything ourselves and each and every piece of content has to be moderated. I think this is like crucial to Openbase.

[00:35:33] JM: What aspect of the engineering are you focused on today?

[00:35:37] LG: So we're mainly focused right now on extending the product, A, for like additional languages and frameworks. As I mentioned Python and Go and other languages and be basically deepening the amount of like insights and content and useful data that we provide for our users. So in that sense we're like highly, highly focused on the web development side. And that means frontend and backend. On the frontend like we have a lot of challenges around the world of like building this sophisticated user experience with a lot of charting, but it still has to be performant. A lot of information architecture. We use GraphQL to so like interact rapidly between the backend and the frontend. We have a lot of challenges on the backend because like we already have tons of data that we're dealing with and we're working on adding like additional interpretations, getting data from like additional data sources. So a lot, a lot, a lot of like backend challenges for like probably the foreseeable future.

So our focus is mostly on the WebStack right now. We're actually hiring frontend, backend, like full stack developers. And over time I would guess probably in a year or two we're going to make a shift and probably allocate even more resources to things like machine learning and big data, because we do accumulate huge amounts of data. And I think at a certain point it would be inevitable for us to like start using machine learning to improve our algorithms mostly around like the discovery element of Openbase. Being able to surface like interesting insights, interesting packages, interesting updates that are like specific to you as a user based on like your previous activity on Openbase.

[00:37:32] JM: Well, Lior, is there anything else you'd like to share or discuss about Openbase?

[00:37:36] LG: I would just say that like we're growing as a company. We're a small startup. We're based in the bay Area. I think we've built something that a lot of developers love and that really makes us happy. And we've just been part of YC's Summer 2020 batch, which has been an amazing experience, and we have backing by great some like high-profile investors from the industry. And going forward, like we're trying to expand our team. So like we're looking for great people who love working on developer tools. We're looking for frontend, backend engineers, full stack engineers, and we're actually hiring our first developer advocate. I think that would be a super interesting role. Like being a developer advocate for a product that is actually so developer-centric. I think that is quite a challenge. So yeah, we would love to hear from you. You can always email me at lior@openbase.io. And yeah, thanks for having me, Jeff.

[00:38:39] JM: Thanks for coming on, Lior.

[00:38:41] LG: Thank you.

[END]