EPISODE 1285

[INTRODUCTION]

[00:00:00] JM: browser automation is a deep space. There are plenty of tools like Diffbot, and Zapier, and many others, text scrapers, web scrapers, API's for the semantic web, different ways of structuring the web. And there's also the large category of tools known as RPA, or robotic process automation. All of these tools hint at the idea that the web is basically this big database, and you want to run queries against it. But because it's such a complex structure of a database, you need robots who know how to query it. You need API's that know how to query it.

And Axiom is a system of browser automation. Axiom is a tool that's trying to combine some of the elements of RPA with elements of browser automation. And they're creating a novel tool. I'm pretty interested in it. And I think you will be too. It's a YC company. It's called Axiom. It's really cool. And let's listen to today's show.

Our first book is coming soon. *Move Fast* is a book about how Facebook builds software. It comes out July 6, and it's something we're pretty proud of. We've spent about two and a half years on this book. And it's been a great exploration of how one of the most successful companies in the world builds software. In the process of writing *Move Fast*, I was reinforced with regard to the idea that I want to build a software company. And I have a new idea that I'm starting to build. The difference between this company and the previous software companies that I've started is I need to let go of some of the responsibilities of Software Engineering Daily. We're going to be starting to transition to having more voices on Software Engineering Daily. And in the long run, I think this will be much better for the business, because we'll have a deeper, more diverse voice about what the world of software entails.

If you are interested in becoming a host, please email me, jeff@softwareengineeringdaily.com. This is a paid opportunity. And it's also a great opportunity for learning, and access, and growing your personal brand. Speaking of personal brand, we are starting a YouTube channel as well. We'll start to air choice interviews that we've done in-person at a studio. And these are high-

quality videos that we're going to be uploading to YouTube. And you can subscribe to those videos at YouTube and find the Software Daily YouTube channel.

Thank you for listening. Thank you for reading. I hope you check out Move Fast. And very soon, thanks for watching Software Daily.

[INTERVIEW]

[00:02:59] JM: Yaseer, welcome to the show.

[00:03:01] YS: How are you doing, Jeff?

[00:03:03] KM: I'm doing well. You run Axiom, which is a browser automation platform. And browser automation is near and dear to my heart, because I think the second internship I had, I was doing automated testing in a system. Maybe you've heard of it, called Watir, which is Web Application Testing in Ruby. And essentially just you write scripts in Ruby. It opens up a browser. And you automate doing stuff. And in doing this show, I have found that web application automation is a large domain, and it's a domain that never really goes away. It's one of these recurrent domains. So I'd love to start with what are you doing in the world of browser automation that has not been done before?

[00:03:58] YS: That's a great question. So you're right, it's quite a hard problem actually. Anybody who's written those kinds of scripts that you've written, there are very similar kind of frameworks in Selenium, which you may have heard of, and Google's Puppeteer, which Axiom is actually based on. But anybody who's written those will find that it's actually a non-trivial problem. Usually, what ends up happening is the maintenance of those scripts is really quite a pain. You will spend a lot of time like having to rewrite them when the page structure changes, and so forth.

So we've done several things. The first thing that's different, I guess, is to make the whole thing no-code. That's actually a very hard problem. Because when you have a developer writing scripts, they can specify the behavior exactly. Like when they click on a button, they'll know to say, “Click on the button with this label or this name.” When it's a non-code that’s specifying it,

basically the tool needs to be smart enough to capture the person's intent. So that's a much harder problem than it might sound. The second thing is dealing with the DOM, the browser DOM. Anybody who's ever dealt with It knows it's a very hostile environment for automations. So you could say we do a lot of the heavy lifting. But the long story short is the primary thing is it's a no-code tool. And as a secondary thing is we do a lot of heavy lifting to take away the work from even developers.

[00:05:16] JM: Tell me about the product development process. How do you find a set of unique niches in the world of browser automation, which is so thoroughly explored? How do you find enough of a market to build a successful business around?

[00:05:34] YS: So, one of the best ways to do it is actually just an empirical approach. So a little bit of background. Before we got into Axiom, we ran a software consultancy, and we often did work in something called RPA, which is actually what you might call Axiom is, its robotic process automation. The kind of knishes we were looking at before very much kind of enterprise oriented. And when we released Axiom after our previous company, we opened up to everyone, not just enterprises, to SMEs. Like all across the spectrum. And all we did was basically just observe what was happening. And that proved to be quite interesting, because the kinds of use cases we thought people would use it for versus what actually ends up happening were quite different. We thought that it would end up – The main niche would end up being things in like, I don't know, like sales and marketing, because that was what our initial ideas were around. Like the kinds of things people do in LinkedIn. What we actually found is those kinds of niches, they aren't very important to people.

In contrast, if something is a recurring and essential business process, it turns out to be a lot more important. And like, for example, we see this in ecommerce actually, and we just discovered this completely empirically just by putting the product out there. Within ecommerce, there's lots of admin work. A very good example is if you're an SME selling on Amazon, or Shopify, or Magento, a lot of these SMEs are copy pasting data from that platform into a fulfillment partner’s website. There's a whole lot of this kind of copy pasting admin stuff in ecommerce. And that's been a really interesting niche for us.

The second niche, which has sort of emerged organically, startups who say we provide an API to do X, where X is something in the real world. There're plenty of API startups. And most often, the thing that they need to write an API for doesn't have an action. So they need to interface with the real websites. And we find they're actually our most exciting Axiom users, because they're building the whole business on top of Axiom in some cases. But those two nieces in particular proved quite fruitful for us so far.

[00:07:33] JM: If I go to your website, I see a wide variety of tasks that you can help automate. So report generation, data entry, web scraping, connecting to other services, clicking buttons, and links. So you can very quickly get to a sense of something similar to an RPA platform. So this robotic process automation, we've done some shows on this. Tell me about the market for robotic process automation, and how that differs from what you're trying to do.

[00:08:09] YS: Yeah, that's a great question, because robotic process automation has become quite a developed market over the last five years in particular. And the 800-pound gorilla in the room is a company called UiPath, who arguably pioneered the space. RPA is actually quite an old technology. It's been around since the early 2000s. And although developers are probably familiar with terms becoming buzzwords and marketing terms, RPA has become one of them. What it classically means is you just automate the user interface. And that's how UiPath started off actually, as a testing tool, which automate the user interface. What they did is they took it to the enterprise sector, and they found a whole load of new niches there, like, okay, processing invoices in shared service centers for the fortune 500 is one.

What you'd say these kinds of companies do is they provide RPA for the enterprise, for the fortune 500. And they provide it as a developer tool, which is most often sold via consulting, like via Accenture. Even our last company basically was doing that. It was doing software consulting, integrating these kinds of providers. So what Axiom did is something slightly different and that we aren't just going for the fortune 500. In fact, we're intentionally going for the longer tail of the market, not just the top end, everything from SMEs, to scale ups in particular. So that's one thing.

The second thing is it's not designed just to be a developer tool. Developers do use Axiom to shortcut what they're doing. But it is actually designed to be an automation tool for anyone.

UiPath, Automation Anywhere, BluePrism, these are all RPA tools that are targeted at developers and consultants, whereas that's not really how Axiom is built. It's designed as a no-code tool. But long story short, is it's RPA for everyone. It's for every size business, and it's for non-coders to.

[00:09:56] JM: One of the more interesting plays on this market that I've seen his Diffbot. Have you seen Diffbot?

[00:10:05] YS: Yes, I've seen similar tools to this. These like monitoring services, right?

[00:10:09] JM: Well, Diffbot – No. You may have it confused with something else. But Diffbot, basically what they do is they scrape the internet, or they index the Internet. And they're a semantic web company. So they turn every web page into something that is more structured. They take all these structured web pages and try to bring structure to it. And in the structure, their structured format is used as the API endpoint. So one thing I'd like to ask you is the process of building a web scraping API, it strikes me as something that's very subject to sites that are built weird or structured weird in unusual fashion. How do you normalize for those kinds of weird data shapes?

[00:10:56] YS: So, what these services do, they have to make a set of really broad assumptions. Axiom doesn't really do that. If you want to turn a website into an API with Axiom, you'll need to build a no-code bot to go and get the data that you need. So it's a bit more like building something and a lot more structured. So hypothetically speaking, you would – Let's say if you wanted to do – I could get the data for a price comparison for a competitor, which is what some people do with Axiom. They build a set of bots that know and understand the competitors' forms, and will know what to click on to get certain outputs from those forms. And each one of those bots will need to be built individually for every different competitor that you have. So the approach is slightly different, whereas those are really sort of generic indexing services. Axiom is more of like a no-code tool to automate the steps that you would be trying to do anyway so you'd specify what the explicit clicking and typing events are.

That also means that it can work behind logins and the like. I get the impression with those kind of structured Indexing API operations. They aren't designed to run what – Their core use case

isn't necessarily designed to run behind the login and go and perform operations, whereas that's Axiom's core use case. If you're an Amazon seller, it will log into an Amazon account. We don't store your credentials up. I could explain that in a separate point. But it'll log into an Amazon account. It will go and execute maybe the complex workflow that you do within the Amazon account, like the steps of clicking and typing. Those indexing services wouldn't necessarily do that, because, they don't have access necessarily to everything behind logins. And it's a slightly different kind of use case.

[00:12:40] JM: Gotcha. What are the domains, the customer use cases that you're seeing the most of?

[00:12:49] YS: So, like I mentioned, ecommerce seems to be a big one, at least as far as the unit economics go that are good for us. There's a whole bunch of other ones that have occurred that we've decided –

[00:13:00] JM: Sorry to pause you there. So when we talk about ecommerce, what is an ecommerce application of a web scraping system?

[00:13:07] YS: So, firstly, I would say we're not necessarily just a web scraping system. We're a process automation system, right? So web scraping just means you get data. We automate a process, which might mean you get that data and you put it somewhere else, right? So within the ecommerce context, that is, if you're an Amazon, Shopify, or Magento store, or one of these major platforms, you may have to connect to external services. You could do that via Zapier if the API's exist. But if they don't, right, now, you're left to admin.

What Axiom will do is log into your data within Amazon, extract that data, then go and log in to maybe your fulfillment partner's website and input that data into the webform. So it's not only the extraction. It's the loading. So some people call that ETL, extract, transform, load. That's actually our core use case more so than just scraping. Should I step back a little and then talk about the other use cases beyond ecommerce?

[00:14:03] JM: Actually, go a little bit deeper, please. ETL – I want to understand what you mean by ETL in this context.

[00:14:09] YS: Yeah, sure. So just to step back a little bit more abstract, you could say Zapier is a kind of web application glue, right? So, Zapier aims to glue together any two web applications that have API's. So, an example, Zapier use cases, every time you get a file sent to your Gmail, send it to your Dropbox, right? Send that file to your Dropbox. So that's kind of an example of ETL with API's. What Axiom does is does that gluing when API's don't exist. So there's a very long tail of web applications that either don't have API's, or the API's are incomplete.

So what usually happens is there're people gluing together these applications by moving data from one system to another. I look at it kind of as a little bit more – Look at it a little bit more abstract, but whereas Zapier is kind of the glue to glue together applications and transform them when there are API's. We do ETL. We load data between applications when there isn't an API. Does that make sense? Or was it a little bit too abstract? I could give a few more examples.

[00:15:15] JM: Yeah, let's do a few more examples.

[00:15:16] YS: Yeah. So a classic thing might be government data websites. So we see this a lot with startups. Many government websites don't have API's. So if you wanted to find out information about a company, if you wanted to do a series of things using government data, you currently don't have much of an option. Axiom could login to that government website, get data on a company and plug it into your system. There're a few other ones that we see, for example – Food delivery seems to be quite a big area actually. If you're trying to consolidate lots of data from different delivery providers, or if you're trying to – How can I put this? Update data like menus and stuff on some platforms. We see this quite often with Axiom as well, where people are using Axiom as the backend to automate, I'd say, menu updating on a platform that doesn't have an API to update the menu.

We see this in financial services where there's – Okay, in Europe, you have open banking, which means there are API's into banks. In Africa, in Asia, in Latin America, you don't actually have API's into banks. So if you want to build a FinTech, you have no choice but to use RPA or use people in the backend. And we actually have some Fintechs doing some things in Axiom that we didn't anticipate. They're actually using it to log in to bank accounts to get statement

balances, to initiate transactions, where there isn't an API. They're doing it on their own system. It's a little bit more secure. They take ownership of it.

But it's this case where, okay, in Europe, you have an API into banks. In Africa, you don't. So if you want to build a FinTech kind of company, you either have to like do a bunch of manual stuff or use a tool like Axiom. Does that kind of example make sense?

[00:17:06] JM: Definitely. I think it's interesting now to talk about the product construction. So you have all these different API's, all these different systems that you've built, it does kind of look a little bit like Zapier in a certain way. What's your process for building and maintaining the wide variety of little services that you have?

[00:17:30] YS: Most of the time, actually, we don't necessarily build everything ourselves. What tends to happen is a customer – Like if you go to our website now, you'll see a whole bunch of templates. And these actually came from our customers using Axiom, and we asked their permission to turn them into reusable templates for other people. What we think will happen with tools like Axiom in the future is that rather than going to develop a to maintain like a set of like brittle Python scripts, if there's any aspect of the application that's brittle or needs to change, we think people on the ground can do it relatively easily. And now we do have what we think are reasonably clever algorithms that can capture like page structure changes, but it won't be able to capture everything. Luckily, Axiom are sensitive enough and easy enough to use no code tool that if the page structure changes, it's relatively trivial for a person to go in and reselect them.

What we've basically seen in general is if there's a person that's responsible now for an admin task, it's kind of like the Axiom bot is sort of tied to them. So they're responsible potentially for maintaining it. So it's kind of like we don't see as like taking away that person's job. It's more like a multiplier for the work they can do.

Yeah. I mean, in the future, what we quite like to do is to like incentivize people to build and make their own. We've actually had a few like RPA consultants come in and say like, “I want to like build and make bots for customers and resell them.” And so that's potentially another layer that we can introduce into Axiom, like kind of a marketplace where if someone wants to build and maintain bots for other people, they can do that. And that's kind of actually a proven model

for the rest of RPA. That's how a lot of the larger companies have worked. So yeah, I mean, the long story short is we see our customers using on the ground. And we see in future like consultants and developers also maintaining their own in a marketplace fashion.

[00:19:16] JM: So, the marketplace, this is not something you have today. This is just a vision for the future?

[00:19:22] YS: We actually do have a set of templates on our site now. It's just that these templates aren't being produced via a kind of a marketplace dynamic, where people can self-publish. Right now, like we curate them and we put things up there. In the future in order for the market – So you can say the only difference between what's there now and what's there in the future is, in the future, you will be able to self-publish. Right now what happens is we curate them.

[00:19:46] JM: It's kind of a cool idea. So that's sort of like have you seen rapid API?

[00:19:52] YS: No. I haven't. No.

[00:19:54] JM: Rapid API is like a marketplace for API's. You're thinking a little bit more specific. About browser automation API's.

[00:20:04] YS: Yeah, yeah. I mean, I guess in some instances we do both browser automation and we link in with other API's. Yeah, kind of like I think IFTTT has kind of a marketplace if you've seen that kind of thing, where people build and publish IFTTT automations.

[00:20:21] JM: Yeah.

[00:20:23] YS: IFTTT is a little bit more b2c, I think.

[00:20:26] JM: Yeah. So when I actually execute one of these API's, where is it running? Is it running on a Lambda function? Or is it running on like some Kubernetes stuff? What do you got going on behind the scenes?

[00:20:36] YS: That's a good question. So there's two ways that can run – Actually, maybe you could even say three ways, that can run on your local machine, in which case we don't touch or process any of your data. So a lot of our customers are dealing with sensitive data. It will only run things that way. And we only store the script, the outlines, the steps for the execution, differentiating between code and data. So that's one mode. You can run it on your local desktop. The second thing is you can run it on our infrastructure on – It's built on Kubernetes, or at least this version that's being released now is. So you can do that too. In which case, we would end up having to process your data, which not everybody wants to do. And the advantage of that, though, is that it can run on a cloud on a schedule. It can be triggered via Zapier. It can be triggered by our own API's.

And the third way to run Axiom is on your own infrastructure, where we can take our cloud solution deployment infrastructure. That means it has the advantage then of being able to run 24/7. You would be processing your own data. That's a good question. I mean, a lot of it just depends on you can choose where you want to run it. And most often that choice is dictated by people's data privacy concerns.

[00:21:45] JM: How are the economics for this business? And how do you price API calls?

[00:21:51] YS: Right now, we've done consumptive pricing, which may not be the right strategy in the long run. This is consumptive in the sense that we charge for successful runs, not failed runs. Because one thing, if you've ever written a script, you'll know is that not everything that attempt to run the thing will work. So that's actually how we launched.

What we've introduced now, though, is we've started to gate some of the features. For example, whether it's scheduling or running in the cloud. Answering your question, right now we're doing consumptive pricing based on successful runs, but we're moving to more of a value-based pricing method gated by particular features. But people that want to run things in the cloud, this starts to become – It starts to become a little bit easier, because there's a whole set of features people normally want to do once they run things in the cloud, like, I don't know, concurrency. And they sometimes have things about – If they have requirements about uptime, for example, if it's a very important application, it's going from consumptive to value-based pricing basically.

[00:22:50] JM: Do you expect the services to be continually these sorts of lower level commands, like clicking links and buttons on a web page, selecting dropdowns, web hooks, that kind of stuff? Or do you expect, like kind of higher level automation? Like, in the future, navigate through an entire website and index the entire website? I could imagine that being a bot service that you would want.

[00:23:18] YS: Yeah. I mean, the smaller automations really are how people begin with Axiom. They usually like a gateway thing. What we usually find is it takes time to build up to the larger ones that they can do in stages. You could say the smaller-based automations are actually more like the kinds of things people might do as individual productivity tools. What we usually find is like a business requirement is slightly different. For example, the kind of Fintechs and sure tech that using Axiom have much more complex automations that they want to run. Quite often, there's very complex decision logic, like, try this bot. And if it fails, try to run it on again under different conditions. Link in to another bot if this occurs.

I would say the long story short is if you're using Axiom for individual productivity tasks, you'll have very small automations. But if your business has a workflow that it's automating, you're doing much more complex automations that are also like linked together and potentially conditional on other things. So, yeah, there's kind of a difference between like b2c-like use cases and very much b2b complex series of bots use cases.

[00:24:28] JM: If I look at the demo, the use of your automation UI, the user is basically sequencing a number of steps. So you have like a task where you might grab a list of links from pages. You have a set of interactions with those pages that you grab. You have this essentially sequence of steps where one depends on the previous step. How do you manage the data from one step to another? How do you manage the dependency graph of the different steps?

[00:25:07] YS: Yeah, that's a great question. So like we have different kinds of step, and between the steps we have like a sort of a way that you can move your data. You have like a little draw that you can see, “Okay, this was the data that we grabbed earlier.”

Usually, the person, whoever's using Axiom, if they're a coder, they'll usually get the concepts quite quickly. They’ll usually understand that, “Okay, the data that I got in step one, that's a

variable. And I can get that variable and it's in a space that I can then access in subsequent steps." Non-coders don't necessarily grok those concepts quite so easily. So we usually have to like give them a bit of training. We might use different words like token to drop that variable in context.

Just going back to the concept about the kind of step as well. So like if you have a variable that you get into the first step, maybe you get not just one data point, but you get multiple data points. In subsequent steps, we'll present you with options to loop through those points. So it will intelligently guess that, "Okay, if in step one, I've got a whole bunch of data from a page. In step two, I might want to then loop through that data. So it will create like a looping construct. Most often, the looping construct is like interact with a page.

So to go back to the example, step one, I might go to a page and get data, in which case I've got 10 data items. And then in step two, I'll make the suggestion, "Okay, maybe now I want to interact with the page." And you can then select like a form input. And it will automatically loop through – If you pass that into step two, it will automatically know to loop through that form 10 times to input the data in 10 times. It's actually a really good question. It really boils down to like creating like a very small programming environment for people, one that needs to be intuitive. If I'm perfectly honest, I don't think we've got there, but we've taken some steps towards that. And our next iteration should be a little bit further. It's basically a mini-programming language at that point.

[00:27:00] JM: So what is the interface between the different steps? Can you just describe the programmability in a little bit more detail?

[00:27:10] YS: Yeah. So if you've ever used Zapier's builder, you might be familiar with this. So what it does is you have like each step is kind of context dependent. So if you've like got data in step one, it'll make different suggestions for step two to utilize that data. And the first step will – If you go into Axiom's builder, the first thing it'll do is give you some example templates to say, "Okay, maybe you want to try and do this. If you don't want to use one of those templates, you can try building from scratch," in which case it will suggest the most common first steps. The usual first steps involve getting data. So that could be getting data from a page. That could be reading data. If that isn't one of the ones you want to use, it will select other things.

After you've, let's say, got data in step one, it will try to make an intelligent guess about what you want to do with it in step two. So you click on like a plus button. And then you come up with another suggestion to say like, “Okay, now that you've got data, maybe you want to interact with a page, or maybe you want to write that data to a Google Sheet, for example,” which you might want to do in a scraper.

The Long story short is it's very similar to Zapier’s builder, and each step – As you click plus and try to add on a new step, it will try to make a guess about what you would like to do with it. And if it's doing its job correctly, it tries to make an intelligent guess that is close to what you want to do. If it hasn't done its job correctly, when you click plus and select a new step, you might find a little bit of – You might have to do a little bit of looking around and try and find the step that you need. I describe it as like an intelligent builder that will try to guess a heuristic for what the next step is.

[00:28:50] JM: Okay, zooming out to the business, a competitive landscape. If I look at the recipes page of your website, recipes sound a lot like things that I would do by stringing together Zapier commands. So, for example, if I want to automate DMing a bunch of Instagram users about their latest post, there's a recipe for doing this. So I've always thought Zapier was like a huge opportunity and that there were other alternative approaches that a Zapier-like business could take. And in some ways it looks like you've built a Zapier-like business. I'd like to know, how are you thinking about the business directionally? How are you think about differentiating from Zapier? Or do you want to be a lot like Zapier? Give me your competitive framing.

[00:29:46] YS: Yeah. So the main thing to understand between Axiom and Zapier is, basically, Zapier is API automation. So you need to have an API. Axiom is RPA automation. So you automate with user interface. We do have API's that we can interact with to, like triggers Zapier. To frame what we are relative as Zapier, I would actually say we're a superset. We can automate any task that Zapier can automate, because we can call it, and we can do more, right? So, in a sense, it's like it's a very much Zapier’s business model, but trying to cover a broader universe of tasks.

The second thing that's slightly different about it is we actually distribute as a Chrome extension. So most apps aren't necessarily tied to individuals, whereas Axiom being a Chrome extension is with you throughout your working day. Whereas if you have a zap, it will go off and run on Zapier servers and not interact with the person. If you have an Axiom, it can then hand back to the person to make a decision. And we kind of see this a lot. I mean, people call this – In RPA, they call this attended automation, when human and machine work together. So like we see this kind of in ecommerce where people are doing admin work processing returns. And they may be automate one bit, and then they hand it back to the person to make a judgment call before carrying on another few steps of automations. So that's another way that I would say that we're a superset of Zapier. We can do not just unattended automations, which is what Zapier does, but attended automations that require a human being in the middle.

But very much in terms of the templates, very much in terms of the distribution model, like it's a freemium distribution model, very much in terms of the kind of persona that we're after, because Zapier has a lot of overlap with like Airtable, Bubble, this kind of no-code tool thing. We are very similar to this Zapier. It's just, I would argue, we're really just a superset of that business in many respects.

[00:31:43] JM: Interesting, a superset. So what are the kinds of API's that you would develop that would not be in the scope of Zapier? Can you go a little bit deeper on that?

[00:31:54] YS: Yeah, sure. So Zapier requires that website to have an API, okay? So if you have, let's say, okay, a very old web application, that's probably one of the classic examples. You have a very old web application from many years ago. Let's say a company Internet system. You won't be able to build a zap for that, because there's no API into that like old legacy system. But you will be able to build an axiom for it. I mean, you'd be surprised how many like kind of ancient systems there are in like insurance and a few other sectors. So yeah, if you have a person who's doing like a bunch of admin work with one of these old systems, it would never be possible to build a zap for that. Whereas with Axiom, that person doing the repetitive admin work can record their clicks and their steps to enter data into the forms and then turn that into recurring automation basically. I think if you want to visualize a good example, it's anytime you have an old legacy application would be a good way to think about it.

[00:32:56] JM: What's the long term vision for the business? What is this business look like in its biggest form?

[00:33:03] YS: I think in its biggest form, if you want to just zoom out a little, rather than just get into the nitty gritty of what we're doing today. If you think about it, the web browser, it's a passive medium. You spend your whole working day in the web browser, and you're doing tasks. Some of these tasks are repetitive, right? There's this capacity to actively automate what people are doing by having them instruct their browser, say, do this and turn that into an automation, right? And that's basically what we do now.

There's definitely the capacity. People are trying this. Nobody's ever really succeeded. There's definitely the capacity to also like passively help people automate by, let's say, observing what people do. And they call this process mining, by the way. By observing what people do and then potentially offering either suggestions, like maybe you should try automating this. Or at the highest level, to try and actually auto generate the automations. If you think about it, like all of the stuff that people do day-to-day kind of goes to waste as far as the data goes. It's only really used towards ad tracking, right? In theory, your web browser could be used for more than just ad tracking. It could be used to automate more of the work that you do.

I think in its biggest form, that's quite a huge opportunity, because there are billions of web browsers. And there're billions of people who are currently inputting useful information to their web browser about what work they're doing. And that could be used more – It can be used for a better purpose than just sending app data to Facebook and Google. It could be used for you.

[00:34:39] JM: Do you have any other futuristic ideas about how web automation could be used, new kinds of applications that could be built or are being built?

[00:34:50] YS: I'm not sure about new kinds of applications as such. I think one of the more interesting things is to look at it as a bit like a new kind of – I look at it as a new kind of programming language more than anything else, because it's basically a user interface-based programming language in a sense. So like you can look at it like something that's a way to automate that's accessible to more and more people. Like you basically automate by mirroring the real actions that you do every day. As for the new kind of application, I think what that really

– It's more about turning the average person into this –People call this idea, this citizen automator. This person on the ground that's able to automate things without going to a developer. That's really how we see that, see this going.

I guess in terms of the applications, I mean, you can look at the startup use cases that I mentioned earlier, and because we do have startups using Axiom. Usually what they're trying to do is they need to automate something that doesn't have an API. So they're doing something that's a new kind of business. I'm trying to think of an example like that doesn't break someone's privacy concerns. I mean, the financial examples are a good one. There're certain kinds of FinTech that you really just can't build now in Africa, because there's no API, and whereas these FinTechs can now use Axiom to build that kind of business and to automate something without having to have people on the backend doing things. I know that's a bit of a roundabout answer, but that's how we kind of see the longer term.

[00:36:16] JM: That's actually a great answer. You must know like what the early Plaid product was, right?

[00:36:24] YS: I don't know about the early product actually. No. That'd be interesting to hear.

[00:36:27] JM: Oh, yeah. If I know the story correctly, the first version of Plaid was basically – You know what Plaid does, right? The banking API thing?

[00:36:38] YS: Yeah. Yeah.

[00:36:39] JM: Basically, if you're going to hit one of these backend Plaid endpoints, what's happening is like a VM is being spun up and it's just logging in – Sorry. A headless browser is being spun up. And it's just logging in to your bank. And it's scraping the bank, like scraping the banking website, because you can't build a banking API when Plaid got started. So they just built headless browser infrastructure that logged in and manually gathered the information from the banking website that needed to be done. I mean, your description of how to build FinTech with this kind of scraping or browser automation infrastructure, I mean, that's makes complete sense.

[00:37:21] YS: Yeah, that's literally what people are doing with Axiom now in those territories where there aren't API's. And it's the same thing for, I guess, any other kind of operation that doesn't have that. Yeah, we've seen some interesting stuff like – I don't know, we've seen like, in ecommerce, for example, we've seen merchandise manufacturing done this way as well. So like if you've ever seen like Figma or in Canva, for example, right? There's a bunch of people that produce designs based on templates. And what they're trying to do is sort of systematize these designs made to order by using bots basically. We see this quite often where you have this thing being spun up that sits behind something when someone produces an order. So, yeah, it basically enables new kinds of businesses to a degree.

[00:38:07] JM: Cool. Well, look, it's been a real pleasure talking to you. I think you got a really cool business. And it'll be fun to watch it in the near future. Anything else you want to add, Yaseer?

[00:38:07] YS: No. Just, yeah, thanks for your time, Jeff. It's been great talking to you.

[00:38:20] JM: Thank you. Wonderful.

[END]