

EPISODE 1129

[INTRODUCTION]

[00:00:00] JM: Robotic process automation involves the scripting and automation of highly repeatable tasks. RPA tools such as UiPath paved the way for a newer wave of automation, including the Robot Framework, which is an open source system for RPA.

Antii Karjalainen is the CEO of Robocorp, a company that provides an RPA tool suite for developers. Antii joins the show to talk through the definition of RPA, common RPA tasks and what he's building with Robocorp.

[INTERVIEW]

[00:00:35] JM: Antii, welcome to the show.

[00:00:37] AK: Thank you.

[00:00:38] JM: Let's start with a simple question. What is robotic process automation?

[00:00:42] AK: Yeah, that's a good question to start with. Robotic process automation actually is about automating business processes through kind of normal user interfaces. So, getting rid of repeating monotonous tasks basically. That's the simple answer, I think.

[00:01:00] JM: Well, describe some of those tasks, the problems which RPA solves.

[00:01:03] AK: Well, RPA can be used in so many different ways. Essentially, anything that you can document, you can automate basically with RPA. You can think about like customer service tasks, setting up new customer accounts, or deleting old ones, creating new bank accounts and so forth. But then again, it can be used to solve more IT-related tasks, like troubleshooting. Let's say you have a cloud app that goes down and you get a monitoring alert. You can set up a robot with RPA to troubleshoot some of that.

Essentially, any task that is repeatable enough so that you can describe it, map it out in a process description. You can go and automate it. And I think one of the key things to highlight in RPA is that it's always sort of agent-based. So you have this software robot as we call them that will perform those tasks. Obviously, not every task is – Even though it can be done with RPA, it doesn't mean that you should do it with RPA. But wherever you need this kind of agent-based approach for automation, that's a good fit for RPA.

[00:02:07] JM: Can you explain in more detail that example you gave where your server goes down and you use the RPA tool to boot it back up. Why would you do that? I mean, it sounds like just this problem for a script to solve or some kind of Terraform thing.

[00:02:22] AK: Yeah, sure. I mean, that's the case if you don't have the need, don't use RPA. But let's say you have an application that's running on your private cloud somewhere and you need to, let's say, validate that it's functioning accordingly, you can think about it like your typical test automation use case in that scenario. In fact, a lot of RPA tools have been evolved from test automation tooling. You can think of it as almost setting up like an end-to-end test in that scenario.

[00:02:53] JM: What are some of the popular tools for performing RPA and how do they vary from one another?

[00:03:01] AK: So, RPA has started – These tools have been started already a long time ago. So I think the market leader is now called UiPath, and they have various others. I think right now there're more RPA tools out there than ever before. There's kind of a boom in them. And how they differ, you said, the traditional approach to RPA is based on this programming model that you use visual interface to build the automation. So you drop boxes and draw arrows between them and set it up.

So most of them actually use Windows workflow foundation to do that as the baseline. But that's kind of the traditional RPA approach, and there has been kind of this idea in the industry that anyone could do it essentially. That even business users could start using these RPA tools and start automating their own work. But how the industry has evolved, it turns out that RPA isn't

actually that simple to pull off at least in more sophisticated use cases. So you tend to have dedicated RPA developers who work with building automation.

So our approach at Robocorp is to actually build more developer-oriented tools, kind of fully on developers tools for people who build RPA as their living. So that's kind of the big difference in the kind of old generation and what we're doing.

[00:04:24] JM: What does that mean? Developers tools around RPA?

[00:04:28] AK: So, the kind of the core of what we do I based on an open source project called Robo Framework, and that's a Python-based tool. You can create Robo Framework strips with just a regular IDE or any text editor for that matter. We support VS Code through an extension that's coming up shortly. Then we have a dedicated [inaudible 00:04:51] environment built with JupyterLab that's custom built for RPA use cases. So it's really good for iterating through various automation tasks and find-tuning them. It has a dedicated user interface for finding UI element locators, etc.

[00:05:08] JM: And that varies from the UI path or the traditional RPA tools and that those require more of a human-based training approach, right?

[00:05:20] AK: Yeah. I mean, the traditional approach is to use just kind of visual representation of a process and automate it through that. But I see kind of many parallels with that approach that we had in, let's say, C++ software in the early 2000s. You used to have these UML-based applications that you define a UML diagram to essentially code an application. They were kind of popular for a while, but never really took off. And I think that's a good reason why this didn't take off in that time, because it turns out that people – We have a really good set of tooling for writing code and working with code. So, why try to figure out some other new way of essentially writing code for RPA specifically? Why is RPA somehow different? So we just go with code instead. It's obviously a code that's obviously built for automation tasks. But nevertheless, is text form.

[00:06:17] JM: Now, I used to use a tool called WATIR, web application testing in Ruby. And this was a thing where I would write scripts to, for example, do automated testing on a forum or

text boxes and enter in basically QA testing. And you could script-up QA testing. How does that differ from what you're talking about?

[00:06:40] AK: It's actually strikingly similar on the technical level. This leads up to the backstory behind Robocorp and what we're doing. I had a background with this open source project called Robo Framework, which is essentially a test automation framework written in Python. Open-sourced in 2008. It's widely used. Very kind of generic and key-word driven. Pretty solid too and has a stable user base.

When I first saw RPA, I wondered exactly like how is that any different from software testing? QA testing? How come is RPA different? When you look at it from the surface, you kind of see these tools, these companies advertising things like AI and robots doing stuff by themselves, and self-learning and so forth. But when you actually go like on the technical level, you'd realize that, "Wait a second. This is exactly the same thing as you would see in QA testing typically."

The technologies that you used to access these interfaces are exactly the same. Where you have difference is, is actually on the management layer of this robot. You have different kind of use cases and RPA that you don't bump in testing world. And that's where it starts to differ. And so you need to manage these scripts differently that don't live in a CI server somewhere and run like over and over again the same thing. In RPA's space, it's actually a problem when something fails. In testing, you kind of want that thing to fail. So that many difference is when you get into the nuances, but on the bottom level in how you automate something, that's strikingly similar.

[00:08:22] JM: And looking at what you've built, it also looks like the interface is a little bit different. It's more of a declarative interface rather than like a scripted interface, right?

[00:08:30] AK: Yeah. You typically write things like open browser. Navigate to some page and download a document and open the document, read an Excel file. Send a bunch of email, etc., these kinds of things.

[00:08:45] JM: So it is still imperative. It's just higher level.

[00:08:47] AK: Yeah. Robo Framework has these keywords that causes those tasks that perform the automation. So it consumes this keyword syntax, and they are essentially functions. But you write them in a way that kind of makes sense in a human language way. When you start automating something, you kind of write a story. So you describe it like to a summer trainee essentially. And then you break those sentences into these keywords that perform some tasks. And those kind of break into sub-tasks and those kind of break into Python functions underneath.

[00:09:22] JM: Let's talk at a higher level for a little while longer. So who is using an RPA tool in an organization?

[00:09:31] AK: That's a good question. Kind of the hypotheses has been that your average business user would use an RPA tool. But like I said, what's happening in the real-world is that you'll have dedicated teams of RPA developers who build that stuff. And smaller companies might use an external service provider who maintains RPA for them. So you have people who offer this as a service-based model.

I'll come to your company, automate your payroll, monthly payroll processing and maintain it for you for a couple hundred bucks for months. And you end up receiving kind of the automated results of that work. So that's one model. But then, yeah, it's dedicated RPA developers.

[00:10:13] JM: Payroll automation. That's interesting, I guess, because like if I go into ADP, ADP is like a really complicated interface and it's kind of this old legacy interface thing. I could easily see wanting to script something over that and make it higher level.

[00:10:28] AK: Yeah, definitely. I think the first task that I automated for myself was logging my hours into Oracle NetSuite when I was working at the other company. So I didn't want to go through NetSuite at the end of each week myself. So I just scripted that and automate it away. There's a lot of these kind of sort of low-impact things that are around everywhere. And then there are like really high-impact things, like let's say financial institutions where you have to go through like millions of records and some people are actually doing it by hand as their job.

[00:11:08] JM: Could you define the requirements for building an RPA tool in more detail? What is this kind of tool? What is this kind of platform have to satisfy?

[00:11:19] AK: So we need to have, firstly, the developer tools obviously. So something that you use to build. For us, it's those – Whether it's VS Code or JupyterLab-based IDE. And then you need to have this kind of interfaces to different underlying technologies. So you need to be able to interact with the browser. There are many good tools for that, like Selenium. And then you need to be able to interact with, let's say, desktop applications. You might need to be able to do image recognition based things for Citrix connections or some applications that are difficult to instrument otherwise. So you'll have an image recognition based component that can handle that.

So the list goes on. So you constantly come up with new things where you need to integrate to or new more convenient building blocks for your users. That's the developer side tool. Then you go into package management and these kinds of things. We leverage the kind of normal developer ecosystem. Obviously, we recommend using Git for package management or handling sort of team work and this kind of thing and not package this necessarily.

And then another large component is going to be the way you actually operate and orchestrate, run those robots. So when you develop something, how do you share it? How do you make it run in a company? You need to schedule it. If it's something that runs by itself, or then you need to distribute it to users if it runs in, as we call, attended manner, if it runs as an assistant on your computer. So those are a bit different use cases. So this orchestrator is a platform that allows you to distribute your robots and operate them in either as an assistant model in the background in kind of a back office and do this.

This orchestrator platform is actually quite complex and has a ton of features that are pretty common in the industry. The ability to manage work queues for the robots, pass along data between robots. Secrets that the robots are going to use, because you don't obviously want to store any password or usernames in the robot's source code, and the list goes on.

[00:13:32] JM: So the open source platform advantage seems to be because there's a lot of domain-specific robotic process automation scripts you might want to write, like things in GitHub

are all similar. Things in Trello are similar. Things in ADP are all similar, and you can imagine wanting a tool suite for each of these different verticals.

[00:13:55] AK: Yeah. I mean, all of those could be driven through APIs or just the browser. So the way we look at it is that you kind of cover kind of the major technologies that way. So API-based automation, that's given, then browser, then desktop applications. At some point, you want to stop building, let's say, Salesforce libraries, or NetSuite libraries. Some really big applications that a lot of people use. So it might make sense to combine multiple technologies like APIs and the browser and build something cohesive for Salesforce specifically. And with Robo Framework, the benefit there is that it has been for over a decade in testing. So you have this vast amount of different technology and domain-specific libraries so that people have it shared and passed along.

[00:14:45] JM: Okay. As we're talking about an individual RPA task, what is the input and output? What can we expect from a typical RPA task?

[00:14:53] AK: You can define input as a user. So you might give a customer number that it needs process, or a document, or something like that. It doesn't need to do so. But typically, you'd want to give some specific information like that as an input. And then a single task can – So you could imagine do all sorts of things that it has been told to do. And the output might be data that it passes along to another robot. It might send an email. It might give a notification. It might just pass and succeed down silently. It really varies according to your specific need.

[00:15:31] JM: Okay. So Robocorp as we've described is an RPA tool suite for developers. Can you say more about how that target customer, the developer being the customer. How does that change what you want out of an RPA tool?

[00:15:46] AK: Yeah. That kind of changes a lot of things. We don't emphasize as much the kind of ease of use aspect. This kind of sounds weird, but we don't try to play down what it's like to build an RPA task. It can be easy. It's not like rocket science or anything. But there are some complex things that people are doing with these tools. So we want to give them tools that are powerful and have a lot of expression. So the tools that don't get into your way, but instead tools that allow you to combine them in ways that we couldn't even imagine.

Just as an example, I think, today, one developer in the community shared a project that he had done, and he had integrated Robo Framework with Bugsnag service for logging errors and keeping a track on robots' daily tasks. And that's something that we didn't kind of – We hadn't have been able to implement that kind of functional yet inside our cloud. But he'd gone and used an external service to provide it. That was pretty cool to see that our tools are flexible enough for people to start doing things that we didn't think of. That's really one key area where I think it differs. We don't have this unified messaging that everyone should become an RPA developer as some of the other companies out there.

[00:17:11] JM: How do I describe an RPA task in Robocorp?

[00:17:15] AK: How do you define an RPA task? You set up a new project. Let's say you're using the lab. You create a new project. You'll import a few libraries. Depending on what kind of task you want to do or what kind of technologies you want to interact with and write `***tasks***` and then start writing instructions.

You might use building keywords, some library keywords that you have available, or you might define your own keywords. So let's say you start with open browser. Obviously, opens the browser. Then you start defining your building keyword. Log in to NetSuite. And then it goes on from there. That keyword should then probably type in a URL or navigate to a URL and do some logging tasks. That might lead you to use world, which is in the cloud, in a cloud service that can provide you secure storage of user credentials and you'll use another library to access the world and so forth. That's kind of the basic process.

[00:18:21] JM: Can you give a few more examples of tasks?

[00:18:23] AK: There are so many types of different tasks. I usually say that there are kind of Snowflakes. Each company, each user has their own particular need. One company actually was the company that had done the Bugsnag integration. They used RPA to automate the process of doing online store purchases in Japanese websites. That was a good use case I think. So they are kind of the typical financial institution use cases that you could imagine that,

let's say, you need to update a lot of customer request at once and people are calling in on the phone and updating, let's say, their phone number for your record.

You might use RPA to navigate through your internal applications that you use for customer records and go through, let's say, multiple locations where you need to update the client information. That's one use case that I saw. And it was actually done in so great volume that the bank that did it said that they saved what it was like 7 years of customer waiting time on the phone by automating that process.

[00:19:33] JM: Very impressive. And when a task is processing and it fails in the middle, what happens?

[00:19:40] AK: Yeah. Typically, you probably want to retry. So just get the input that you have and retry it if that's how you configure it. You probably want to notify somebody. So you'll send out an email or alert through some other server desk application. And depending on – Obviously, it kind of depends on why the failure happens. Was it a business exception? The application that you were automating did something wrong or then your data was incomplete that you used, or something like that. So that kind of guides you. If it's just a flat-out failure, just exception uncaught, then you probably want to get the developer to look at it.

[00:20:22] JM: Does something happen like when the UI of the website changes? That can sometimes mess things up? Or do most of these sites tend to keep their UI static enough that your tasks don't go out of date?

[00:20:34] AK: Yeah. There's a bit of art to that. So you want to try to use locators that are more stable. If you use like Absolute, XPath references, those might get messed up pretty easily. If you use like Element.id, that's more stable. So typically – Yeah, sure. I mean, sites changing. That's an issue for RPA. You can do something as a developer to make it more robust. But ultimately if your site ends up changing completely, there's nothing that you can do about it. Typically, I tell people that if you have the option, just use an API instead, because that's more stable. Always go where you have the most stable route.

[00:21:17] JM: Tell me more about the libraries that are built into Robocorp.

[00:21:21] AK: Yes. We are actually developing a fairly large library that we call the RPA framework, and that consists of your basic toolkit essentially that you need as an RPA developer. So you'll have things for the browser, things for desktop applications, image recognition. You'll have integrations with all the major cloud platforms, AWS, Azure, Google Cloud. And probably forget like 80% of the stuff. We keep adding to it like multiple times a week. We'd come off with a new release that adds new functionality to the RPA framework right now at the moment. But yeah, it's a multitude of different kind of keywords that you categorize by different technology domains.

[00:22:09] JM: There are lots of APIs out there for machine learning. Tell me about how machine learning APIs could be used with Robocorp.

[00:22:17] AK: Yeah. The industry term for that is intelligent automation. And you typically call RPA intelligent as you hook into some machine learning API. But, sure. I mean, kind of the basic, I think most frequent use case is to do something like send a document to AWS Textract or the similar Google Cloud Vision API. You have, let's say, a PDF invoice and you want to extract some data out of it. You can easily use this kind of cloud services to do that for you if you don't want to try to do it locally. That's I think the most common use case that I see all the time.

[00:22:54] JM: When an RPA task gets spun up, what is actually happening on Robocorp?

[00:23:00] AK: Yeah. When an RPA task gets spun up, kind of if you look at the architecture that we have with the cloud platform, we have what we call the workers. The worker is essentially an application that's installed in the target system. So it can be on your laptop. It can be on a virtual machine something. Or it can leave inside a container that we can host for you. Depending on where the worker is.

But let's say that the worker is on my laptop. What happens is that the cloud will send a package of code and some instructions to the worker. The worker will get the package, unpack it and then initiate a fresh conda environment for the Python libraries that you have inside your robot's code and then it'll kind of start fresh every time to make sure that you don't have any side effects from previous executions.

And then when you have your environment done, it'll execute the task, stream the consult. Tries it to your cloud account. And then when it's done, it's going to stream your execution artifacts. You might have produced some documents. Or you at last have a log file. You'll stream both to the cloud and report there, resolves the execution.

A lot of the stuff that we provide is really the ease and convenience of just you want to run Python-based Robo Framework automation. Good. Just install one app in the target environment login and you're good to go. You're all set. You'll have a stable execution environment each time. And it's pretty fast too.

[00:24:37] JM: What are you using under the hood to orchestrate these tasks?

[00:24:40] AK: The orchestration service on our Robocorp cloud is we are working actually mostly serverless. So it's on AWS. We are not using any ready project to set up the orchestration. The scheduling on those, we do it ourselves. And then it's a fast-growing platform of different features. It's one of the core pieces that we develop.

[00:25:05] JM: Well, can you tell me more about that? The serverless orchestration stuff? How much can you offload? What kind of leverage do you get by going serverless? What does that look like?

[00:25:15] AK: We started developing that early 2019. And at that time, we made the decision to kind of go as cutting edge as we could, because we have the opportunity. So my VP of engineering just made the decision that if we don't have to host anything by ourselves, we won't do it then. We might have a few service here and there, but most places, it's 100% serverless. The idea there is that we should be able to scale up pretty nicely with that decision. Obviously, you're going to always have some issues here and there. But for the most part, we think that that will allow us to maintain certain level of service as we continue to grow and scale. And, really, some of these processes are pretty business-critical to companies. So we need to be careful with the operations.

[00:26:05] JM: Tell me more. I just want to know more about your infrastructure. And like how are you using Lambda? Are there any other serverless services that you'd like to discuss?

[00:26:14] AK: Well, I mean, it's a typical host. But, really, I'm not too deep into architecting our cloud service. So it's basically like SQS, DynamoDB, Lambda all around. We use [inaudible 00:26:26] for monitoring, Datadog as well, I think. Yeah, it's a complex and growing piece of software.

[00:26:34] JM: What programming languages do you use?

[00:26:37] AK: Mostly, JavaScript, Typescript.

[00:26:40] JM: Are there any other cloud services you use? You mentioned Datadog. Anything else?

[00:26:44] AK: For the SaaS apps that we use, I think we are getting segment now into the mix. I can't remember. There's a long list of this. But trying to figure out which are particular for the product side or the operations side. And we use Linear for backlogs and these kinds of things. Kind of fairly standard sort of SaaS stack altogether. The company is pretty young, so we've gotten to choose kind of the best that are out there right now at the time.

[00:27:15] JM: What has been the hardest engineering problem to solve?

[00:27:18] AK: I can't name a sort of one particular, because we actually have – Yeah. I think the hardest part has been kind of to manage how big our product is. Essentially, we have to cover a lot of ground. So we are developing three things at the same time. We have our learning hub, learning resource side, Robocop hub. We want to get user-generated content there as well in the mx. So we have that going. It's the simplest part technically. But then we have the developer tools. We have JupyterLab that has its own ecosystem and kind of own conventions and everything related. That, we're developing on it. Then we have VS Code that we support. Command line tools at the same time. And then all the [inaudible 00:28:04] that we had developed on top of it.

And then we are also active in many open source contributions. We recently came up with a language server implementation for our framework. And now we're building a proper Robo Framework debugger as well at the same time. So that's one side. And then we have the cloud platform as well and that we're going to distribute across. And the cloud is just a large product as itself. I think managing all of that. Trying to keep some sort of focus. That's the biggest challenge.

[00:28:37] JM: Yeah, you've got a really wide platform at this point.

[00:28:40] AK: Yeah.

[00:28:41] JM: Is it challenging to maintain all that stuff? Are there part of the platform that get out of date that require continual maintenance or do you feel like you're more consistently building new things?

[00:28:54] AK: I'd say we are building new things. It's not like we would have to stay on the sort of latest and greatest with some sort of technology. So we don't need to follow that much. Obviously, as thing progress – Things like [inaudible 00:29:09] automation and that kind of things, they really don't move that fast and leverage other projects. So the Selenium project is maintained in the web drivers, for instance. We benefit from that.

And now we're actually coming out with a new browser automation framework altogether based on Microsoft's playwright. And that's exciting to follow. We kind of leverage a lot of things that happen around us at the same time. And we also are lucky to get some community contributors as well. That's starting to happen, which is exciting. So I kind of highlighted our developer tools. Everything that we build under the developer tool's umbrella, that's open source. So we benefit from a lot of things from that side.

[00:29:53] JM: Tell me about the companies that use Robocorp. What problems have they solved?

[00:29:58] AK: Right now, we have to distinguish between using Robocorp and Robo Framework. So, Robocorp is fairly new to the game. So we started 2019 and we got funded by

the kind of towards the end of 2019. So we've been able to ramp-up some of the operations just last December and January. We just came out with general availability of our products on July 1st, and we are going out with paid offering on October 1st now.

So there's a limited set of real-world projects that I've seen. So I don't know if I've seen every project that's out. But then there're kind of companies that have already prior to Robocorp starting using Robo Framework for RPA. I was kind of initially involved in pushing Robo Framework, the project on the open source side to become an RPA tool. I know many companies that have already adapted it by themselves and built tooling around it.

But now for Robocorp, I'm seeing things that are built, let's say, for automating tasks in healthcare sector in the US. One company in the Netherlands, they were doing automation for general practitioners for doctors' offices. And we don't usually go into super close details with our users when they have client projects, because they might have limits to what they can disclose. But we try to keep tabs on a lot of them.

[00:31:23] JM: Tell me more about the community for Robocorp and the surrounding open source projects.

[00:31:29] AK: Yeah. Robo Framework itself, actually, the project started at Nokia early 2000s and was open-sourced in 2008. And in 2015, we created what's called the Robo Framework Foundation. It's a nonprofit organization that supports the project financially. We have, I think, close to 50-member companies there that contribute financially to the development of Robo Framework. I'm actually currently on the board of directors of the foundation. So that's one big community. We have a forum and active Slack workspace, and their core Robo Framework tool gets downloaded around 5 million times annually at the moment.

And then we are building the kind of RPA-specific community inside RoboCorp. So we have a Slack workspace as well and people are sharing their ideas and asking for help and providing this cool project that they are doing on the side there. That's one thing that has been really exciting to follow. Just being there and watching how the RPA-specific community has started to grow and thrive. And kind of the first people inside that community have been this kind of early

adapter, smaller boutique consultant companies that I'm personally kind of been following that trend.

So you have these people who might know about software engineering know about RPA automation. And then they discover Robocorp and say that they could actually make a career for themselves in building stuff for others without tools. This sense of enthusiasm as people are finding in them.

[00:33:08] JM: How do you expect the RPA world to change over the next 5 to 10 years?

[00:33:12] AK: 10 years is a long time. 5 years is also quite long to predict. But I say that open source RPA technologies like Robocorp, they will definitely change the business model. So we are taking the business model actually to consumption-based model at Robocorp. It used to be so that you'd buy these licenses for Robots. Essentially, one execution environment that is allowed to run an automation task. That would count as a robot, and you'd pay like \$15,000 annually, something ridiculous [inaudible 00:33:47] for the privilege of being able to run it.

We are coming out with the consumption-based model where you actually pay for the execution time of the robot in a sense, rather than committing to a huge investments upfront. So I say that that model is going to bring up RPA to kind of down market from the large enterprises. We are actually starting – We see ourselves starting like bottom-up. So going to smaller companies [inaudible 00:34:13] instead of like we don't really too much enterprise stuff right now. So that's going to be a trend.

Then kind of looking at the largest perspective, we have Microsoft entering the RPA space quite recently with an acquisition. So that can be something that changes the landscape quite a lot depending on how they play it out. I just say that the use of RPA is going to go more and more mainstream. People are going to discover more opportunities out there. It's going to become widely accepted as you start having these smaller players who will offer it as a service to other companies that you just hire somebody to automate your accounts receivable or what have you.

[00:34:58] JM: Do you have any other predictions for the future and things around software development and how software development might integrate with RPA more intimately?

[00:35:07] AK: Yeah. I mean, I'm pretty pragmatic in that way that I don't generally encourage people to use RPA where that's not necessarily. But I do think that there are interesting use cases for agent-based automation that we haven't fully discovered yet. And one thing that I'm following closely that I'm seeing in my email inbox is kind of request as people want to integrate RPA into their products. So people say that they have, let's say, a SaaS application, and they could actually benefit from being able to provide their users with RPA-like functionality. So we have been extending our cloud API that allows companies to do that.

And I think that that can actually be sort of a bigger change in how you, as a developer, somebody building a product would perceive RPA. If you're able to use a set of convenient APIs to integrate software almost into your own product offering.

[00:36:09] JM: What about the intersection of RPA and higher level or no-code applications?

[00:36:17] AK: No-code, or then you have low-code sometimes. It's a good question. What I'm seeing is that you have a ton of this workflow automation tools, sort of Zapier and these kind of things that you can use to automate. Like you have an API call that provides some data and then you transform the data and call another API, that kind of thing. Those are useful and will continue to exist. Those won't replace RPA though, because you often want to have specific tasks, kind of agent perspective and the ability to go into your local infrastructure. As I mentioned in the beginning, you can have this robot act inside your private cloud or even on your own laptop. You can have that with sort of online workflow automation tools too easily.

But obviously you can have that if you use our cloud API, and you can use our service to integrate that way. So kind of RPA, I say that it's still continuous to be its different thing. In many cases where you have this kind of fantasy where you would be able to build up processes without writing any code, you end up kind of bumping into models limitations really easily. And you need to have a sort of expression power of code to cover all your bases in RPA. It becomes pretty complex pretty fast. And then you're in a low-code tool and try to write like C# in a dialogue box somewhere inside of a visual tool. Really not optimal.

[00:37:52] JM: So, we could talk a little bit more about the interaction pattern with Robocorp. I think we kind of glossed over this. If I'm writing a task in RPA or an RPA task in the robot framework, what is my experience in the IDE? What am I actually doing?

[00:38:11] AK: Yeah. That would be typically you're writing keywords and finding UI locators. So you would use this UI locator tool there to inspect your application that you're automating, finding the relevant locators and then using them while you write your keywords. So click a button, navigate this side and so forth. That's kind of a typical workflow. We try to make it easy and sort of fluent.

You might have workflows where you do something on the web for instance where you perform some action, like navigate somewhere, fill in some forms or whatever, and you record those actions into keywords. Often, we have a lot of record and playback test tools, but they've never gained too much popularity, because the end result become pretty brittle. Even if you use something like that, you need to go through it and use your own judgment where you need to modify the script and where you can improve it in some ways. As I mentioned, there's bit of art to building RPA. How you structure your robot task and how you structure your locators and so forth. But that's kind of the main activity there. Being able to inspect your application, describing your actions that you need to perform in the application and then iterating and then iterating. And we tried to make the iteration cycle as easy as possible inside our developer tools obviously.

[00:39:38] JM: All right. Well, to close off, what have you learned about management in management a large team, division of labor? I mean, you're the CEO of this company. I'd love to know more about what you've learned.

[00:39:48] AK: Yeah. I mean, a lot of things obviously over the course of this one and a half years that we've been running the company. I started early 2018. So I've been at this for many years already even before the company was formed. But we went from 6 people to like 35 into a matter of months after the funding, because we knew that we have so much things to build. Right now, we are at 42 people, I guess. And we are spread out remotely. So, almost as a fully distributed team. So that layers in kind of one additional aspect that we need to consider.

I think there's no particular thing that I can highlight by sort of a million smaller management insight. I'm trying to constantly read management books and use our board as an asset to learn from them and use other people around me as an asset to learn is going to be a constant journey.

[00:40:49] JM: Antii, thanks for coming on the show. It's been great talking to you.

[00:40:52] AK: Thank you.

[END]