EPISODE 1336

[INTRODUCTION]

[00:00:00] KP: Instabase is a technology platform for building automation solutions. Users deploy it onto their own infrastructure and can leverage the tools offered by the platform to build complex workflows for handling tasks like income verification, and claims processing.

In this episode, we interview Anant Bhardwaj, founder of Instabase. He describes Insta base as an operating system. We explore what he means by that and discuss the types of use cases Instabase powers.

[INTERVIEW]

[00:00:32] KP: But other than that, Anant, welcome to Software Engineering Daily.

[00:00:35] AB: Thank you so much. It's great to be here.

[00:00:37] KP: We brought you on to talk specifically about Instabase, so let's jump right into it. What is Instabase?

[00:00:44] AB: So, Instabase helps large enterprises, like large banks or insurance company, your health care or government automate their most complex business processes. So just to put into perspective, what does it really mean, we work with companies across financial services, insurance, healthcare, and so on for the kind of problems like let's say, your Goldman Sachs and you want to automate how to do lending. Whenever somebody applies for a loan, you should be able to give that loan in less than five second. That requires rethinking of the entire architecture. Like how do you get the documents? How do you automatically understand them? How do you automatically apply all the business logic? And then being able to do all that into a solution that can help with this entire business process, Instabase provides a platform to do all of that.

[00:01:38] KP: Very cool. Well, I can imagine a lot of complications in that process of getting the five-minute loan approval done. They have to probably upload some documents that who knows what those look like, right? There are so many different pay stubs in this kind of thing. Maybe there needs to be OCR and problems like that. Do you consider yourself an AI platform in that regard?

[00:01:58] AB: So, AI is one important aspect of it, because how do you really understand document? You have to apply AI. How do you really classify certain things? You have to apply AI. But there are a bunch of other pieces that are beyond AI, which is like, how do you make an API call to some credit system that will give you credit score, right? How do you really integrate with some already fraud information that you might have about that particular account or that particular individual. So, the amount of AI that are used in some aspects, which is document understanding, extracts, and so on, but then what we call this like Flow Builder, which is a low code platform for building all the business logic in addition to the AI, because the good part of the systems are also a ton of business logic. If you know, any complex business process, there is so much of those.

[00:02:52] KP: Absolutely. Is Flow Builder, the main tool that an operator of Instabase would find themselves in most of the time?

[00:02:59] AB: For automating their business workflows. Flow is one of the main tools, yes.

[00:03:02] KP: And then the typical implementation or deployment, how complex does Flow become?

[00:03:08] AB: So, flow is basically a graph, where you have steps that can branch out and you have more steps that you can build as complex as you want. So, it's basically unbounded graph. You have set up steps, and those steps can connect to each other in any way. So, it could be very simple, like, do the set of five things, or it could be very complex, like, go do this based on this decision, then do this, then go and apply AI here, then based on the AI result, do some other things then do 20 other things. So, it depicts and represents the complexity of the business process. And Flow is a tool that allows you to represent all that complexity pretty easily.

[00:03:49] KP: If I didn't know about Instabase, and I was tackling a problem like this, I'd probably go hire some software engineers who know about API integrations and maybe PDF processing tools and things like that. In this situation, would I ask that same developer to use Instabase? Or do you find a different type of professional is better to operate a tool like yours?

[00:04:08] AB: Yeah, this is a really, really good question. So, if Instabase didn't exist, I'm sure you can build number of different pieces using developer but there is a lot of work that is needed to really solve some of the problems in this complex business processes. That's why you see a bunch of these business processes are very manual. The reason why when you apply for a VDI, it takes a long time. The reason why when you apply for home loan, it takes a long time. Because a bunch of these things have to be reviewed by humans, because machine or AI has not reached to the point where they could make these decisions automatically.

Over the last two or three years or so, there is a significant advancement in artificial intelligence and deep learning that allows the human level precision for these kinds of problems. So that's one aspect. So many times, that would be developers at our customers who will go and build that business workflow using the toolkits and building blocks we provide. But in many cases, the ops people because they understand the business problem much better than developer. Since we already have a very visual tool to be able to build that workflow, so many simple workflows can be built by non-technical ops people themselves. But if you go in very complex workflows, then developers can go in because if you want to integrate with API, some developers will have to write that integration code.

[00:05:26] KP: If we were implementing these sorts of things in software, I assume we'd pick up the typical tools like version control and pull requests for software deployment. What does that experience of change management look like in Instabase?

[00:05:39] AB: Yeah, that's a great question. Because in Instabase, you can write code, you can write flows. So, we have version control built in the platform itself that integrates with your GitHub, and Bitbucket and GitLab and so on. Because different customers have different kind of version control system. So, what we do is we basically integrate with the existing version control software, so that they can start versioning the artifacts that are created by Instabase.

[00:06:02] KP: I'm wondering if we could do a zoom in on the flow system and talk a little bit about the options available there. If users haven't been there to check it out themselves, yet, I think they could picture this being an audio podcast, maybe it's a little challenging, but picture kind of a palette of different tools like we'd mentioned extracting stuff from PDFs. You probably have a library of those and the graph you can connect things with, how big is that palette? What's the universe of options look like to a new user?

[00:06:29] AB: Yeah, so you have to basically think at the raw level and the meta level. So, at the raw level, you might have infinite options, because you can do whatever you want. But the meta level, there are not many things. If you see, there have been software that had been created that encode these complex workflow systems with some simple primitives. One of those primitives, like the first operation is you do certain operation that can be repeated by a code. But then when you are working on batch of data, you have some kind of map operation, then you have some kind of reduce operation, then you have some kind of combined operation, then you have some kind of filter operation.

Now, within filter, you might use a different classification model, depending upon your business problem. Within mapper, you might use read file as your map, for every single file, read that and convert this into text, that's one map operation. After reading the file, then you might want to split pages. That could be another map operation. Then you can apply the filter operation that does your classification using some model. Then on the classified file, you can use another map operation that really basically uses some extraction model to extract the fields, and so on.

So, there are some set of like simple primitives, like about 15 or so. And then those can be mapped into your actual business operations. Even though there could be infinite steps, they eventually get reduced into these 14, 15 common primitives that you use for building the workflows. Pretty similar to like, if you used a Spark, right? Like Spark basically gives you some primitives to build these complex data pipelines, and all your like MapReduce combined filter and all that. But now, what goes inside map is the interesting part. Is map like OCR process? Is map like splitting into pages? Is map like making an API call somewhere? Is map like writing custom user defined functions? All the other things that are flexible.

[00:08:13] KP: Gotcha. I've seen a lot of implementations of OCR, and I found it works great, modern tools, even off the shelf stuff work great if you have a narrow range of documents you work with. Someone says, “We want to process these tax lien documents, and they're all basically the same thing.” You can get that going. But Instabase has a much harder challenge than that. Anyone can show up with any crazy PDF that has a terrible layout and this kind of stuff. Are there challenges you face around extracting data from the universe of all possible layouts?

[00:08:47] AB: Yeah. So, this problem was very, very, very hard, as you already know. It was impossible to solve this problem until like three years ago. But the advancement in the AI, so I don't know how much you're aware of the models that Google basically produced called Bert, and then open AI did, GPT, one, two, and three. And what they really did is they took the entire worldwide data, and you could mask a particular set of words, and see that if deep learning can produce the same set of words. Because if it could do that over the World Wide Web, then you can reasonably claim that if accuracy is high enough that it really has ability to understand language.

Now, if that is true, document is superset of language, right? You have all the language problem, and then you have all the layout because just some random number at the top would be invoice number or purchase order number depending upon the document. So, what you really do is, can you take all the documents in the world on a two-dimensional plane, and rather than thinking those as a sequence of words, can you just think of those as something represented on a 2d geometry? Because at the end, what is a paper? Paper is a 2d geometry where you have these words, occupying some space in the geometry. Now, can you build a deep learning model that can take all the documents in the world and you can mask any area on the two-dimensional plane and see that if machine could reproduce that.

We got a really good breakthrough sometime last year, where it was able to do that with a very, very high accuracy, like 80% to 85% accuracy. That's pretty incredible, if you can do that, which basically means that deep learning was able to understand different layouts and the concept of documents, like what is list? What is your paragraph? What is your table of contents? What is selection box and all of those? And then what you do is given that you have that base model, now you can train that for the specific domains. For example, you can now take 100, like pay

stubs and train a model, it basically learned that concept very quickly and it basically gives you very high accuracy in extracting from pay stubs. Same thing you can do for any kind of documents.

So, the reason why this was possible was because of the advancement in the deep learning that happened over the last six, seven years or so. But the key breakthrough was the attention model that Google wrote the paper about, which fundamentally change the natural language that you saw, Bitboard, GPT-2, and GPT-3. We're applying something similar, mainly focused on documents.

[00:11:06] KP: Those tools have all been created after the founding of your company, can you talk a little bit about how you stay relevant with research and bringing in such fresh new ideas into presumably your production system quickly?

[00:11:18] AB: Yeah, so this is such an interesting question that Instabase was designed as an operating system. The architecture of Instabase is – and it came from some of my work at MIT. So, MIT, we were doing a project called datahub. And what datahub was looking at was, if you go back in history, you will see that everybody used to build their own data center. But then AWS and GCP, and, Azure came along and said, “You don't need to build your own data center.” Basically, you can use some primitives like compute and storage and caching and databases, and you can just get up and running your data center. Every data center needs similar things.

The question that at MIT, we were trying to ask was, can you build a layer on top, which is a Windows like operating system that the way like Windows building an operating system where you could build all the SDN primitives that anybody could build application on Windows and that could just run anywhere? What is the layer above the cloud? Should there be an operating system, which has really cool building blocks, and that anybody could build applications on? Now, what applications we would build, we didn't know at that time, right? So, I basically dropped out and spent first two years building the OS. Once we build the OS, you want to build some cool primitives and building blocks. One of the key building blocks that we built was document understanding, because that is such a common problem across every single industry.

Our early version of document understanding was powered by a technique that we had done a lot of research on, called program synthesis. But at the same time, we basically saw that as the variability increases, that particular technique struggles, because the point is, it is very, very hard to really come up with a logic that could understand any layout. We have a focus team that really looks at what is happening in this research area and we, ourselves, do a ton of research in this space. When there are tons of models that came from Google, and then we saw that how the effect of that had been on natural language, the fundamental question is, why it would not work on documents. Now, can you go and train a big neural net on documents, and you start seeing the results and that's how we brought that in.

The reason why we were able to do that was because we've designed everything in an operating system and an operating system, you can keep adding more building blocks, right? If you look at Windows, the reason why Windows has stayed relevant for last, like how many years now, they started in, like 1980, or something. So, the actual Windows operating system, it's like 40 plus years is because Windows did not solve any problem, it just give you a primitives so that you can build applications. And those primitives can keep changing. Those applications can keep changing. So, we don't do much get affected by getting obsolete and that is because maybe one component of our OS can get obsolete, as long as we have the ability to bring the latest innovation immediately faster than anybody else. You can still be number one all the time.

[00:13:56] KP: I see a strong path to adoption for Instabase in companies that aren't necessarily technology companies, things like claims processing and mortgage processing in particular. Not that I know those industries, but they strike me as places where lots of companies could be successful, with surprisingly basic office policies and systems in place. Is that your typical path? Or do you see adoption across a wide set of different enterprise situations?

[00:14:22] AB: It's a very wide set. So, basically, our strategy was this. So, when we started a company, we wanted to win an operating system. If you want to win an operating system, and you know, you have to win the application ecosystem. Now, in order to win the application ecosystem, you need a lot of developers building apps for your platform. There is no reason why a developer would want to build an application for your platform unless you have market distribution. It just doesn't work.

So, it's kind of like a chicken and egg problem because the market has to declare you have victor before even if you start with the step one. We would like, if we want to win in an operating system, we have to first win the market penetration. Now, how do you win the market penetration? Nobody wants to buy an operating system. People buy things that solve their critical problems, critical use cases. So, we were like, now let's go to the key industries and find what are the most important use cases and most important problems that they have so that we can bootstrap the operating system by solving the use cases and pushing the OS as part of the platform and then keep pushing more and more things.

We basically started with banking. Our strategy was to go after top players first, right? Because for example, like in banking, the customer that you want to sell to is Goldman Sachs, JP Morgan Chase, Bank of America, Citi, and those people, right? Because once you do that, you cover 80% of the all of the volume, but also, they are the most complex. If you make it work for them, it will work for everybody else. So, what we did was we basically took a strategy exactly opposite of most other people that rather than trying to go after the smaller players, and startups, we will go after the biggest and most complex companies in the beginning across every single industry. So, we did in banking, we did in insurance, we did in healthcare, and now we are basically trying to do that in government. Our customers are some of these big names that you already know, Goldman Sachs, Bank of America, JP Morgan Chase, MetLife, and so on.

[00:16:08] KP: Could we talk a little bit about the onboarding process? What does it takes to get going with a tool like yours?

[00:16:14] AB: So, there are two steps. One is the installation and deployment. The second one is building the applications. So, most of our customers are large, which I already talked about. That basically means there is a ton of regulations. This data cannot go that displays, the HPII and all that stuff. And that's where our biggest advantage is, since all the applications for Instabase are built on Instabase operating system, as long as you can support running the operating system anywhere, your applications are fully portable. When you build an Android app, you didn't matter which phone you built it for it will work on any phone as long as they have Android operating system.

We basically said, “Yeah, I mean, you can install it on-prem, you can install it inside your own virtual private cloud. It doesn't matter. It's AWS, or GCP, or Azure, or you can run on Instabase.com.” So, most of these complex customers typically choose to install within their on-prem or inside their own cloud VPC. That sometimes, cloud VPC is easier because you know the environment. When it is completely on-prem, sometimes it gets tricky, right? Because what is the network policy and what is the storage engine? You have to do all the integrations. We have seen entire infrastructure up and running in production, sometimes as quickly as a week, and sometimes as long as like three months. The reason why sometimes it takes three months is you go and deploy the customers in and then you figure out that there is some security policy that did not allow network calls, or there is some policy that did not allow connecting database in a way that we do, or you have to use certain protocol that is very internal to them.

So, we had to deal with all those. But now given that we have worked with so many customers, now it is pretty seamless, so anybody can go and really deploy anywhere where they want and advantage that customers get is application that they build is future proof. You build on-prem, but when you move to the cloud, same applications will work. When you move from AWS to Azure, or GCP, things will continue to work. So, any application that is built on Instabase is fully portable, fully future proof, it doesn’t matter where you run.

[00:18:14] KP: I imagine that portability is in part, powered by the strength of containerization is sort of a tool to do it. Are there any extra challenges that make it especially difficult to deploy into all the major clouds and even on-prem environments?

[00:18:28] AB: So yeah, it is powered primarily by containerization. But you have to know this, which is containerization gives you the ability to run, compute in a stateless way. But you still have to store the state somewhere, some database, some caching and so on. So, we had to define like, how do you build a general-purpose file system? If you get Windows, the reason why Windows is portable, or you can install on any hardware is because somebody wrote NTFS. And all the hardware provider will basically implement a device driver that basically converts the NTFS calls into that lock storage calls on that device, right? So, we had to basically build a bunch of these very, very basic primitive, like how file systems are written. We have our

own file system called IBFs, which could be mounted to S3 or NFS, or Azure or local file system or whatever.

So, when the application is called Instabase file system, under the hood, it will make the IO calls which are relevant to the underlying device or storage device that actually stores the data. The same thing we had to do for databases. Same thing we had to do for all of the applications, like how the applications communicate with each other. So, containerization is one aspect that allows you to run services in a portable way. But they don't guarantee that every application that you write will be completely portable on top of it, because you have to come up with these primitives that allows these independences.

[00:19:48] KP: Makes sense. Can you talk a little bit about the process of fostering that marketplace of applications? I'm curious about how you get maybe external developers to contribute ones or guide your own internal development for what your users need.

[00:20:02] AB: Yeah. So basically, we break this into three different stages. The first one is the first party developer, which is application developed by us. Second one is second party developer, which is application developed by our customers. And the third one is a third party, which is application developed by somebody other than we and our customers. So, the first part is pretty simple. We build applications. We know everything internal, we can build applications. So, I'll not spend a lot of time on that. So that's pretty cool. That works.

The second party is the application that are built by our customers. And because at the end, like one of the things that we realized, in the first party case, why first party is important is, even though it looks like the document understanding is such a broad market, there is a lesson that we learned, and I'll tell you the lesson. It looks very obvious, but it is hard to realize that until you really sort of see it. We realized that every bank in the US needs to solve same kind of problems. They process similar kind of documents. So like JPMC, they do not process any different kinds of documents than Bank of America, than Goldman Sachs. The reason why, is when you apply for the loan, you will submit same documents to Wells Fargo as you when you submit to JPMC. So, the point is, the problems are pretty much same. Same thing for insurance, right? Same thing for healthcare.

Basically, geography and industry are the most important things like all banks in the US need to do same things. All insurance company in Europe need to do the same things. So, you go after every geography and industry and build useful apps for them. So that's the first party. That's pretty simple.

The second party is slightly complicated. And the reason why they complicated is, like some of our customers are global, right? Like a Standard Chartered for example. They wanted to build an app for client due diligence in Kenya and Uganda and Karna. I mean, we don't know how things work there. They know that much better than us. If we try to go and understand every single market in the world, it will not work. So, giving the right toolkits as part of the operating system so that they can do it themselves becomes very, very important. That's the second party. Our customers building applications for their own internal marketplace.

The third one is why would somebody else want to build applications? So, we went to Google first and say, like, "Hey, Google, you want to build apps for us?" And Google were like, "I mean, why? I mean, we want everyone to come to Google Cloud. You are telling us to build apps for you guys. Why?" Then we went to Microsoft and say, "Hey, Microsoft, you guys are building a lot of cool services, do you want to build and make those things available on Instabase?" And Microsoft was like, "Who are your customers?" And then we said at that time Bank of America and JPMorgan Chase and Goldman Sachs and RBC and First Republic and MetLife, and so on. They were like, this is interesting. And I said, like, what if like, we gave you our services, let's say OCR, or handwriting detection or language translation, can you make it in production in all these customers? They're like, "Sure." And the reason why Microsoft cared about this was because Goldman Sachs went to AWS. Capital One went to AWS. Somebody else is on GCP.

So currently, Microsoft can only capture those customers who are on Azure, right? They cannot capture those customers who are on AWS and GCP. Now, as soon as they have built applications on Instabase, that application is available to everybody. It doesn't matter whether somebody is in AWS or GCP, or somewhere else, or even on-prem. So as soon as they build the application, we saw that, basically everyone started using it, because, you know, it was good. It was a good application, and everyone needed it. Today, I think we drive more volume than any other vendor or partner that they have in terms of the OCR and the handwriting and the natural language translation and those kind of things to Microsoft and this is great for them.

Because even if somebody went to AWS or GCP, Microsoft can still make money from their cloud services, because those services run natively on Instabase, and can run it in the environment of AWS. Because nobody wants to make an API call from GCP, to Microsoft, because you have the security policy and all that.

So, this is basically our third party. So currently, the third party is very limited to selected set of companies. Because when we go to Goldman Sachs and say like, “Hey, this app is built by Microsoft”, they're more willing to use it, than if we say this app is built by some random startup. At some point, we'll open it for everyone, but currently the third party is very restricted to our selected partners.

[00:24:12] KP: Gotcha. That makes sense. When you have a new, maybe even a new industry, take a look at the platform. Is there any phase where you have to go through and kind of sort out unique use cases and introduce new functionality? Or what's, I guess, the maturity of the offering been like over the years?

[00:24:27] AB: I mean, there is always some work you have to do. But those are more market research than technology research. So, market research is what are the use cases? What are the value and those kinds of things? The point is, it depends what those problems are is like, can you run set of steps and can you understand documents at the very core level? Or data in general, can you understand unstructured data better than anybody else? Because most of the problems they have to deal with unstructured data. In healthcare, it's just that your input is going to be a bunch of medical records, and doctor's notes and those kinds of things rather than pay stubs and bank statements and something else.

In general, we do not have to make product changes when we go to new industry, but they always have market research on how the sales team will go and purchasing things and how we will talk about things and so on.

[00:25:16] KP: When a technology group is about to start a Greenfield project, and they're looking at options, and Instabase crosses the table of things to consider, obviously, there's a lot of things to think about along those lines. There'll be some young engineer that says, “No, let's just build it ourselves.” There'll be someone who says, go out and get other comps and other

RFPs and things like that. What is the typical story like for someone who learns about the product, and then their path towards adoption?

[00:25:43] AB: In general, I think some of the problems are very, very hard. So, even those people want to build it themselves. They love Instabase, because in Instabase, they can build that in two weeks, rather than three years. So, it's this question of like, “Is there any sense of building AWS yourself?” It doesn’t make any sense, right? You can go and do the cool stuff on top of it, because it gives you all the primitives. So, I think our goal is to enable all of those developers to go and do things quickly and get a lot more value from their work than trying to build some of those things, which take long time and are so complicated.

[00:26:18] KP: What's the developer experience? If I sit down, and let's just assume I've got something a little bit custom, where I'm going to want to write some of my own code, how do I get to hello world?

[00:26:27] AB: That's pretty simple. We have an ID where you basically open abc.py, put that inside script directory and simply say, “Register hello world, colon, hello world” and write your own implementation of hello world in Python. You can use whatever language you like. But Python is the most popular language that customers use. So, that's for the backend code. And for front end code, we basically have – because many times you want to do customization on the front-end side. So, we have something called custom UI as your widget and you can write any React component there. So, you can basically literally change anything by writing code in React.

[00:26:59] KP: Very cool. The idea of Instabase as an operating system is interesting to me. Operating system is not a framework and I think it's Instabase is not a framework, it's something a little different than that. But it's also a space, I'm not totally – I don't have my sea legs for I guess is one way to put it. Are there other things that you have inspiration from? Or is this a comparable space? Seems like Instabase is somewhat unique in that regard.

[00:27:24] AB: Yeah, there is nobody who is building an operating system. But fundamentally, what we believe is that if you look at Windows or Mac, which has been the primary desktop operating system, before even I go there. So, if you look at how people consume computing,

and I think people consume computing in three different ways. The first one I call lifestyle computing, and lifestyle computing is the computer that you need as part of your lifestyle. For example, maps and camera and search and those kinds of things. And the right operating system and the right computer for that is your phone and iOS and Android, the right operating system.

Now the second one, we call the productivity applications that people can give like Word and Excel, and PowerPoint, and desktop has played an important role in sort of distribution of the applications like Windows and Mac, the two primary ones. So that's individual productivity applications. But more and more of those things are moving away from desktop, right? They're moving towards something on the web site. You go to like slack.com, or you go to Facebook, or instabase.workplace.com, or Google Suite, more and more things are being offered as a browser-based application. So, they don't build native iOS applications anymore.

The third one is operational productivity applications. The applications that allow you to run operational productivity, like how do you run banking? How do you run government? How do you run healthcare? And there's no operating system for that. The question that we fundamentally started asking is like, if you can go in buy an app or getting food delivered to your home on your iPhone, and there are like dozens of apps. Why can't Goldman Sachs go and find an app how to run their risk analysis? It doesn't exist, right? Everything is very custom built, and they have to go and engage with vendors, and so on. And what that OS would look like?

There are two steps that we saw. One was more and more applications, but moving towards cloud. So that's number one. And the second one is the desktop application was moving towards web. Web became the main method of delivery for the applications. There are very few applications that run as a native app on Windows or Mac. Even Slack is basically a WebKit. So, under the hood, it makes a web call. It's basically all of that stuff is contained inside the web layer. If more and more applications are going to be distributed, which is they're not going to be run on single machines. They're going to be running on multiple machines, what that OS would look like? What the distributed operating system would look like? And then we looked at if you have to build distributed application, what is the layer? What is the stack? And the stack is you have some data center, and you have all the cloud services and then you build on top of that. We were like, “What if we built an end user operating system that were distributed? What if we

build an application for Windows, but that could run on tens of thousands machines?" Can automatically scale to whatever things that you needed. And if that would be true, then people can really build the application that can be distributed for real problems that Goldman Sachs or Bank of America or MetLife, or whoever.

So, that was the key hypothesis behind building the operating system. Because once you think of what is the layer above data center, what is the layer above AWS, or GCP, or Azure? And that business logic layer is something that has to be solved by an operating system.

[00:30:27] KP: Absolutely. One of the things an operating system does is offer services, not just services that the applications can use, but ways they can work together in some sense. I could differ operations. You don't get that in the sort of standalone collection of web apps that we have today.

[00:30:44] AB: Exactly.

[00:30:45] KP: What are some of the ways that that manifests in Instabase?

[00:30:48] KP: Yeah, you got a really, really good understanding of what we are trying to basically do. Because the problem is their application, is they can't talk to each other, right? abc.com and xyz.com cannot talk to each other. The reason why iOS is very useful it not because you have camera and GPS, is because you have camera and GPS as a primitive and as a building block, so that you can build your own Uber. You can build your own Instagram. And how you can do that the distributed operating system where these things become building blocks, like deep learning can become a building block, where any applications can go and use that, where basically, each of these applications can communicate to each other through some protocol. Where any application can really use the services provided by other application and they can see what that application is doing. As long as somebody gives you permission. But there is common storage system, right? As long as you know, you have given privileges and permissions.

So, that was the whole region, which is there are so many applications that are being built, but they all are being built in silos where no other application, and there is no way for using the

application as a building block so that you can create the third one. Because one of the quotes that I really liked from I think Steve Jobs, which is, “A platform is not where the whole is equal to some other part, where you can get the value equal to each individual thing. Here, the whole would be like 10,000 times bigger than some of the parts and that's what the good platform is.” You should not see iOS is collection of camera and GPS and map. That's not what makes iOS unique. What makes iOS unique is those things are primitives that allow the applications to be built, where each application can talk to each other and can communicate with each other. And that's exactly why we went after this OS vision.

[00:32:25] KP: Every industry needs to take security and compliance and privacy very seriously. This is especially true in some of the industries you service like banking, and insurance and things like that. What's the story for those topics as it relates to Instabase?

[00:32:40] AB: Yeah, so we have taken more of like Apple's approach towards that, which is, you control the device, right? So, we don't ask customers to send the data to us. We basically say go and install inside your own premises, inside your own firewall, which could be your cloud, which could be your on-prem and so on. And then we guarantee that all the communications and everything is encrypted and we have a very robust permission model, where basically the concept of organizations in the workspaces, and you have these individuals that have access to the workspaces. And then you have physical separation, where it's not even logically separated. Each workspace can mount different storage.

So, for example, I'll give you a very simple example, let's say you're Goldman Sachs, and you have offices in India and UK and US, and you created three workspaces, GSN, GS UK and GS US. It is not sufficient that basically, you have the permission model that says, “Only people within US have access to the US workspace, or people within India have access to India workspace and UK, within UK workplace”, that's not sufficient. Because the regulation says that if you're collecting data in UK, the data physically need to live in UK. If the data you're collecting in US needs to physically live in US. So, the way Instabase does this is you create the logical workspaces. But then as part of the file system, you mount the storage as part of the workspace. So, you don't have this global storage as part of the OS, what you basically say is, now Goldman Sachs India is going to mount the storage unit, which could be NFS, or that S3

bucket hosted in India, that it only accessible within that workspace. But then US can basically mount separate things and same thing you can do for databases and so on.

So, the way we have approached this is by how do you guarantee that there is not just logical isolation and access control? There is ability of physical isolation of data too, because that gives complete control for security, and then this all runs inside the firewall of the customer. So, we don't get to see the data.

[00:34:32] KP: Aha. Well, if I'm deploying your software into my environment, and it's sort of not air gapped, but separated from you, how does billing work?

[00:34:41] AB: It's like, we basically have the sort of system that counts inside that and then we have to go and ask the people, can you give us the report? So, we have that kind of trust. We don't basically try to ask that. We don't try to move the data from their premises to us. So, we literally have to go and quarterly review that.

[00:34:58] KP: Gotcha. Makes sense. What else should developers know about Instabase?

[00:35:03] AB: I think, at some point, and once we open for general purpose developer, like today, if you have a cool idea, you can go and build on iOS and Android app store and becomes available to everybody. Tomorrow, if you have some cool idea, you can build an app, put that on Instabase App Store, and be available to every single large organization in the world, which could be Goldman and Bank of America and MetLife and Standard Chartered and Capital One and RBS, and RBC and so on.

So, that's the key thing. What that enterprise app store looks like? We have a consumer app store, but we don't have an app store, what do you build the application, and you get free distribution, where you can just build massive companies. So eventually, our goal is to enable every single individual or every smart people or great minds in the world, to be able to do cool stuff, and then be able to distribute with Instabase.

[00:35:53] KP: Similar question, but for like an entrepreneur or a product executive at a company, someone who needs to develop a solution, what draws them to Instabase?

[00:36:04] AB: It's all the cool building blocks available that makes your job so easy. It's the same thing at AWS or GCP. Why do you want to build that, is because you don't have to deal with the headache of like scaling, and the storage and the data center and the hardware and all that. You just go and build your services. So here in Instabase, you just go and build your application. That's it. It's automatically we have taken care of distribution, we have taken care of running on any kind of infrastructure, we have best building block. If you want to do deep learning, I mean, if you tried to build your own document understanding, that it will take many, many years. We have put all that effort. So, there are just so many amazingly good primitives, like 150 plus primitives available as part of the OS, that you can quickly build a very, very high value applications and make money.

[00:36:45] KP: What's the future of the company? Are you just an assembly line of new primitives to introduce or is there other functionality that will appear?

[00:36:54] AB: So, we'll continue to build the primitives, which is the OS part. But then we also build application. If you were Microsoft, Microsoft not just builds windows, but they also build Office, right? Word and Excel and PowerPoint, they are real applications. So, we have about like 25, 30 applications too and many of the marketplace applications are built by us. So, we will continue to look at the opportunities available in any single industry. And if those opportunities are, you know, large enough, we will go and build apps ourselves. Currently, like we already have apps in the US markets for all of the common problems like identity and income and any kind of documents related to taxes and all that. The same thing for UK, India and so on. But we will keep an eye on what opportunity to open up. And if there is a big enough market, we will build applications ourselves too, on top of this same operating system that we want other people to build on.

[00:37:41] KP: Makes sense. I suspect someone adopting your solution probably already has some infrastructure things they're doing in their business, probably lots of different databases and tools in play. What's the common story for integrations and pulling things from be it S3 or MySQL, what kind of connectors are available?

[00:38:01] AB: So, S3 and MySQL are more the core OS level primitive. This is the concept of mounting. We already have device drivers that converts into base file system into S3, and so on. So, you can just mount it. All the standard file system and databases, those are just available as part of OS. But if you have something like unique, for example, SAP or Salesforce or something that doesn't need to be OS level primitive, we have this concept of user defined functions, where you can just add the functions to connect to that system, and then convert the data into the format that you want on Instabase.

So, Instabase is a very, very open architecture system. So, anything within Instabase can be called from outside using API, and build in Instabase, you can call any system via API. As long as you can make a network call and can make the other system communicate, you can integrate that with Instabase. Now, if there is no way to communicate with that system, there's nothing you can do so and we provide the protocols for that. We have prebuilt connectors for all the common stuff that you already talked about, all the databases like Oracle, and MySQL, and clouds, and others. And the same thing for file system S3 and Azure Blob Store and Google Cloud Data Store and so on, and NFS, and so on.

[00:39:07] KP: Got it. We didn't get into it yet, very much at all. But I should note you have a PhD from the famous CSAIL at the laboratory at MIT. Can you talk a little bit about how your academic training has influenced your work at Instabase?

[00:39:20] AB: I don't have a PhD because I dropped out. I cannot say that I have a PhD. I was a PhD student.

[00:39:26] KP: Sorry, I misread that. Gotcha.

[00:39:29] AB: But yeah, I spent about a little over three years there. MIT is an amazing institution. I think I'll give a lot of credit to the time that I spent there and the kind of people that I got to work with and the things that I learned from. It's an incredible organization. I think a lot of what we are doing are at Instabase is influenced by the work that I did there. Also, before that, I went to Stanford, and I think I learned there too, a lot.

So, if you ask me, everything that we are doing our Instabase, are the things that I learned at these two institutions.

[00:40:00] KP: Gotcha. Well, Anant, thank you so much for taking the time to come on Software Engineering Daily.

[00:40:05] AB: Thank you so much. This was a lot of fun. And hopefully this was helpful.

[00:40:09] KP: Absolutely. Good conversation. I think we wound well. I want to just touch base to an offer, is there anything you think we should have covered that we didn't get to?

[00:40:17] AB: No. I think you asked all of the good questions and good amount of technical depth. I think, as long as the audience are software engineers, this will work great. If the audience is non-technical, I don't know if many answers would make a lot of sense. But my exception is that audience is technical.

[00:40:27] KP: Absolutely. I think this is perfect.

[END]