SDS PODCAST

EPISODE 996: HOW TO GET $100M RETURNS ON AGENT DEPLOYMENTS, WITH NIKUNJ BAJAJ

Jon Krohn:	00:00	Imagine being able to deploy an AI agent and getting a return of over a hundred million dollars from that single deployment. My guest today has done that multiple times. Welcome to another episode of the Super Data Science Podcast. I'm your host, Jon Krohn. My guest today is Nicunj Bajaj, CEO and co-founder of True Foundry, a Bay Area based startup that has raised over $20 million to solve the thorniest problems that enterprises that organizations in general face when deploying agents and he's done it with great success as I sat at the outset of this episode. He's done this for some of the most demanding organizations in the world, including Nvidia and Siemens. What an incredible guest. What a topic. I'm sure you're going to enjoy it. Nakunj, welcome to the Super Data Science Podcast. How are you doing today?
Nikunj Bajaj:	00:54	Very good. Thank you so much for having me here, Jon.
Jon Krohn:	00:57	Yeah, of course. We're in person in San Francisco in a beautiful, sunny week in San Francisco and here we are indoors.
Nikunj Bajaj:	01:04	How does life get better than that?
Jon Krohn:	01:07	Nice. It's so great to have you on the show. Congratulations on all your recent success. It was really cool doing some research for your episode. Tell us about what True Foundry does and how you've had the success recently, all the capital you've been raising, all the great clients you've been landing.
Nikunj Bajaj:	01:21	Absolutely. I think we owe all our success to agents. Over the last few years, most of our enterprise customers have gone from building agents that are quick prototypes to now things are actually running in production. And as these agents start running in production, enterprises realize that they need a control plane to manage not only these agents, but all the underlying components of these agents. And what we have been building at True Foundry

		is this control plane layer that sits between the application layer and the infrastructure and model layer so that we have one place where you can connect, observe and govern. I like to call it, we are a cog in a wheel, but in a very unique way.
Jon Krohn:	02:06	Tell us a bit about how it actually works. I understand that you have a trillion tokens a day or something like that being processed, which is insane for your clients. Big number. It's hard for me to wrap my head around that. It's a big number. Yeah. Yeah. I don't know how many tokens I hit when I run out of when my Claude code is like, you're out of credits, but I know it's a lot less than a trillion tokens.
	02:27	So yes, a trillion tokens, tons of usage, obviously big clients. I think some of them I can name. I know there are some big ones I can't name, but I know Nvidia is one that I can say. For sure. Siemens is one I can say. And secretly, there's a bunch of other logos that you would definitely recognize listeners that we can't say on air. And I would love to understand, you kind of described the functionality of what True Foundry does, but what is it like as a user to use it? If our listener hears, "Okay, great. I want AI orchestration or security or governance." What do they do? Is it open source or do they go to your website? What's the flow like?
Nikunj Bajaj:	03:06	Very simple. You literally go to our website, truefoundry.com, hit sign up, connect your models, start using it. So let me actually explain how this works from a ... Where do we fit in into a user's journey as well?
Jon Krohn:	03:20	Exactly.
Nikunj Bajaj:	03:21	Imagine a user that's starting to build out an agent. To build that agent, they need three most important things. Number one, they need access to models. And now instead of directly hitting the models, they're going to hit

this through a model gateway layer. They're going to interact with the environment and this environment could be tools, functionalities, skills, actions you may want to take, data you may want to tap into. We call this goes through an environment gateway layer. And in the industry, this is more popular with the term MCP gateway layer. But we think that environment interaction is a little bit more generalized than just accessing MCP servers basically.

Jon Krohn: 04:03 Whereas the MCP servers is for tools alone, but with this environment gateway, it's kind of tools and governance and all the other belts and

Nikunj Bajaj: 04:11 Whistles we offer. Tools, skills, any kind of data that you may want to tap into all of those things. And we think that these are modus operandi today. They could also change in the future. So that's why we think of this systems wise as an environment interaction basically. And then the third thing, an agent will most likely need to connect with other agents or sub-agents and this happens through an agent gateway. So as a user is building out an agent, then interacting with these components, you go through a gateway. And then once all of your traffic starts flowing through a gatewayer, you start also recognizing the fact that now everything is observed. Your developer does not have to instrument their code at all. Everything is governed. It's like entering an office building. You're entering through a gate, you know who is entering, when do they enter.

04:58 If you want someone to be blocked from entering, you could do all that. That's what we are doing for AI.

Jon Krohn: 05:03 I see. So yeah, if you kind of anthropomorphize it, what you're doing with True Foundry is creating a system where you can tell what agent is kind of clocking in for their Workday, when they clock in, when they clock out, what tools they have access to, who else they can speak

to in the office space. I like that. I understand what data they can access and some kind of directive for what they're supposed to be doing.

Nikunj Bajaj: 05:27 Exactly.

Jon Krohn: 05:28 That's cool. And it sounds actually, the way you're describing it, it kind of sounds like it's click and point, even though it's intended for engineers, developers to be using, right?

Nikunj Bajaj: 05:36 Yeah. So yes, we have two sets of users who are directly using it and one major set of persona that's actually managing it. So for sure we have engineers who are building out agents. We call it pro code agents when engineers are building it. They can write their code, but within their code, the API layer, they can invoke the models MCPs and other agents through our gateway. But we also have a UI layer, a playground layer where let's say a person, a user who is not coding on a day-to-day basis like a builder

Jon Krohn: 06:06 Or an IT manager.

Nikunj Bajaj: 06:08 They can actually simply prompt it and it's still everything goes through the gateway. They can build an agent through texting and that agent gets published back to our agent registry so the organization can keep scaling it. And then we have one more persona, we call it the platform engineering team. They're the ones who are in charge of managing the governance and the observability layer within the enterprise and they are the ones who are going to buy our platform, set up our platform for internal usage.

Jon Krohn: 06:32 Nice. I like that. I'm starting to get a really good picture of how all this works. In our research, we came across that there are three specific components to this. So you've talked about the AI gateway, but it sounds like that can

be broken down into these three components, model gateway, MCP gateway and agent gateway. Do you want to tell us about that?

Nikunj Bajaj: 06:51 Yes, for sure. Yes, that's exactly right. So first of all, just a quick industry picture. Sometimes in the industry, people refer to AI gateway as a very specific term, which is basically just a routing layer across different LLMs.

Jon Krohn: 07:07 Like open router.

Nikunj Bajaj: 07:08 Like open router. So to me, that's an LLM proxy layer, a very crucial point in the overall stack of AI, but still like one slice of the solution. We generalize the term AI gateway to truly include AI and all the underlying components of AI, which means the models, the MCPs, the agents. Those are the three major components that we talk about. And we did mention how as an enterprise or as a user, you're never sharing then the direct model API keys to a developer everything, you only give them access to the gateway and the gateway knows which user has access to which models. Same thing. Gateway knows which users have access to which tools, which service, and which agents have access to which other agents. So those three things are going through the gateway layer. And then across all of these three, you get end-to-end observability.

07:56 So imagine this, right? Your agent ended up invoking a model that invoked an MCP server, implemented a certain guardrail that do not send out my PII data to this particular model. Or if there was a prompt injection attack in the prompt, then just block that request altogether. So you can set up all those guardrails and then you have a complete trace of what happened. And then as an enterprise, if you cared about putting certain kind of guardrails, certain kind of cost limits, FinOps layer, rate limiting, all of those things you're able to do through the gateway layer basically. So that makes it a

very powerful interface to build, govern your agents basically.

Jon Krohn: 08:34 You guys have been working on this for years and it sounds like there's a lot of functionality in there. It is interesting how this, it seems like you, I guess, were a visionary around building a platform that all of a sudden then about a year ago with where agentic capabilities landed, all of a sudden all this functionality, things like this gateway you're describing all of a sudden became essential.

Nikunj Bajaj: 09:00 Well, everybody likes to think that they are the visionaries. The reality is that we have been working in the space closely with our customers who have been building these agentic AI applications. And when you care about working with your customer and you're hearing their problems straight and then you're ready to act on it fast, that's when you start looking like a visionary, but truly they were the ones who were doing everything. So the story is that we were working helping our customers build, deploy their agents on our platform from 2023, 2024 timeframe. And we noticed the fact that in the beginning, all of our customers, enterprise customers actually thought of this AI gateway layer as indeed a thin proxy layer and they thought they're going to build it in-house and everybody built it. As they started putting their agents to production, they realized, oops, it's more than a thin proxy layer

Jon Krohn: 09:58 Because

Nikunj Bajaj: 09:58 Now I need to build out this observability layer, this governance layer, this policy engine layer. And now it's not just one team within the enterprise that's working with these models and agents. So many different teams are doing it. All of them have their own cost centers and they could also see that the environment or the ecosystem outside the enterprise was also exploding like MCPs are

coming out, A2A protocol is coming out, new agent building tools like Claude is coming out. All of these tools have access to all of your internal tools and stuff. So they realize that the complexity grew very, very fast. And as we are seeing that our customers are struggling to solve this problem, we continue to expand the capability of our platform in conjunction with them. And here you go, works well for us.

Jon Krohn: 10:46 As the person developing these specific solutions at Truth Foundry, you end up being in a position where you're seeing pain points across multiple different clients in different sectors. And so it gives you an advantage over them. When they see something, they're like, oh yeah, this is something quick I can build, but they're only experiencing their particular, the pain points that they've had so far, whereas you're seeing this much broader range of problems. And so it allows you to make more informed opinionated decisions about how the product should be designed and how it should work, which gives you the edge.

Nikunj Bajaj: 11:21 Absolutely. And you get the best in class from everywhere. This one customer who may have perfect idea of how to solve a certain problem, but they're taking potentially some suboptimal routes to solve a certain other problem. But because we have seen a similar problem with another customer or we have experienced it ourselves, we are able to bring in the best in class everywhere. And actually this is where one of the things that I like to believe is startups at a certain stage have the biggest advantage of this phenomena of getting feedback loop and closing it fast. If you're too small, you actually do not have enough customers to get meaningful data points out of. If you're too big, then people who are making the product decisions are not the ones who are closely interacting with all these customers that are on a day in, day out basis basically.

	12:08	The moment you merge the two, that's when I think magic unleashes.
Jon Krohn:	12:11	Nice. Well, congrats. You're seeingly nailing it right on the head because people are loving your product and flocking in. All right. We were talking about those three kinds of gateways and then I think it took you off on a tangent. We had gotten through the model gateway aspect of it. Tell me about the MCP gateway. So I should also give a little bit of context. Probably most of our listeners know what MCP is, model context protocol, but just a quick primer. In case you don't, it's a very popular protocol similar to the way that HTTP is a protocol for the web. MCP is a protocol for how agents can access tools. And so there are thousands conservatively of what are called MCP servers out there and each MCP server has one or more, sometimes hundreds or thousands of tools in a given MCP server. So if you think about a Google Maps MCP server, it allows you to put in a location and you can get the latitude and longitude out.
	13:10	That's one tool. You could find directions between two points. So all the kinds of things that you can do in the Google Maps app, a agent can do using the Google Maps MCP server. So that MCP server has access to all these different tools. And that what that means is that thousands and thousands of these MCP servers exist and across those thousands of MCP servers, there are millions conservatively of tools that agents could access. So anyway, a bit of context on MCP. So what does the MCP gateway at True Foundry allow you to do?
Nikunj Bajaj:	13:41	Right. Well, first of all, thank you so much for that beautiful explanation of the MCP servers. I might steal that for some of our customers who are starting to get accustomed with these terminologies. The concept of MCP Gateway fundamentally is very similar to the model gateway. Let me actually just explain a little bit more about the model gateway, the problems that it is trying to

solve and then overlay that concept on the MCP server bit as well. So first of all, when you are working with a model gateway layer, you stop circulating the API keys of the different models to the end users, which means now you have a single API key with which you're able to access multiple different models. As an organization, you can put some rate limits on it that you can only invoke this model a hundred times in an hour.

14:27 You can put budget limits on it that in a day, I do not want any dev environment to spend more than $2,000. So you can start doing all that control layers and then you get observability on the model side for free. You could also do interesting things around model fallbacks, for example. What if an open AI model in the US West region was down or slow, you can start routing your traffic to other regions basically. So you can do all of these controls once your traffic flows through the gateway. Something similar happens in the MCP layer, except the authentication in the case of a model is just an API key, but in the case of an MCP, think about it, you're dealing with a lot of other private data. If it's a company's Slack account, Gmail account, whatever it is, it has a lot of sensitive data.

Jon Krohn: 15:15 Yeah. And not only just reading it, but writing to it as well.

Nikunj Bajaj: 15:19 Yes. Right? So now you really need to, and each of these different tools may implement its own authentication authorization mechanism as well. So the problem becomes more complex in the world of MCP. You still want to figure out the authentication problem that who is the user or the agent or any virtual entity trying to access certain MCPs,

15:37 But you also have to solve for the fact that what is the type of authentication mechanism that MCP server is implementing, then you have to create a mapping

between the two. But to this end, I think the point that you mentioned is not just reading but also writing and it's the agents autonomously writing code and invoking these actions basically. So then you also want to have a sense of control that maybe there's an agent which could potentially be controlled by a consumer outside of my organization. There could be prompt injection attacks in there. So by definition, I actually only want to give it read only permissions to like seven different MCP servers. So then you start building out these user interface layers that I'm going to put together these seven MCP servers, but only whitelist that read only tools from these MCP server creates some virtual entity and exposed to this agent.

16:30 So now this agent has read only version of a bunch of my MCP servers. And then of course you want to build out tracking, you want to build out similar kind of resilience, tool selection. All that complexity or harness as people are starting to call it now starts to get baked into the MCP layer of it as well.

Jon Krohn: 16:47 Really cool. Thanks for that tour. I suppose that this tool gateway would also probably help with tools sprawl, which I think is an issue that a lot of organizations face. You can end up in a situation where, because I mentioned it's so easy to hook up via MCP servers so many different tools, you can end up with too many tools being provided to given agents and that can lead to problems. It can dilute the specificity of what an agent is supposed to be doing and it can also potentially be providing read-write access in situations where we might not want it.

Nikunj Bajaj: 17:26 That's right. Yes. So one of the things that happens with the tools Prawl is first of all, fundamentally toolsprawl doesn't break anything. You could have technically a million tools out there and you are individually managing them.

Jon Krohn:	17:40	Yeah. You could have one agent, you're like, "Here's a million tools."
Nikunj Bajaj:	17:43	Figure that out. So there's nothing theoretical about it. It's about the user experience, the developer experience that you provide of how easy it is to manage these agents, who has access to what, and when something goes wrong, how easy it is to debug it basically. That's where I think the productization of this MCP servers and controlling this toolsprawl comes in handy. So one of the examples that I gave you about this virtual read-only MCP server entity that I talked about earlier. That's an example of you know that this agent has to operate in a certain scope. In that scope, they may need access to these 10 tools and you define that to the agent. Now there are also other ways where you do not know ahead of time what are the tools that I want to give access, right? That's where you start creating certain kind of logical entities to manage these tools and you basically create what we call as a description that if the agent knows that this is the segment of tools that I really care about, they can read through that description and then in runtime select these tools.
	18:48	So automated tool selection is the other problem that we solve for where the complexity goes on now from end users to the models themselves basically.
Jon Krohn:	18:56	Nice. Great explanation. So I think we've now covered two of the three kind of component gateways. We've covered the model gateway, we've covered the MCP gateway that leaves us with the agent gateway. How is that different?
Nikunj Bajaj:	19:10	That's the hard part.
Jon Krohn:	19:12	Okay. Okay.
Nikunj Bajaj:	19:12	The agent gateway is the hard part. Well, first of all, remember when I started saying that an agent needs to

talks to models, the environment, and other agents to get things done. Now what's happening right now in the industry is that these agents are taking actions and people are building out agents because they make them look more productive and they're doing things that we sometimes do not want to solve for. And we keep giving access to these agents to bunch of tools, bunch of databases, bunch of actions that they're able to take. What if something goes wrong with the agent? Most of the industry today does not have a kill switch.

19:57 In the best case scenario, they may be able to observe what my agent is doing and that too in the best case scenario, but they do not have a way of stopping it. So to solve for that problem, you want to put your agents behind a gateway. So if something goes wrong, you can manually or automatically kill the agent. Now the chat- So brutal. It's critical, critical, right? It's brutal and critical. However, what's happening is there's a lot of different agent development platforms that are coming up and each of these agent development platforms specialize in their own environment that there could be certain sales tools and they're like, all your sales data is with us and we can give you sales related agents. You may have ITSM tools and you may end up building out ITSM related agents. And then there could be these pro code type of development environments where you're building out agents as well.

20:54 Now the question is, how do you take these agents built into all of these different platforms or frameworks and still put them behind a governance layer as an enterprise? That's the problem that the next version of the gateway will need to solve for. And that's what we call as our agent gateway layer.

Jon Krohn: 21:11 Cool. Right. Thank you for that tour of all of the gateways. So to kind of round that out, the AI gateway is the general product and then within it there are these three

components. The agent gateway that you just talked about, the MCP gateway before that and the model gateway before that. Really cool. So lots of folks listening want to be using agents more often as individuals as well as in organizations from small startups to enterprises. I think most people probably listening to this podcast now have a good idea about how much potential there is with automating workflows and trusting agents to be doing processes autonomously from knowledge worker processes all the way through to automating physical things with robots and so much potential with agents. But there's a lot of trepidation as well because of security issues. And so I was wondering if you could walk us through, it sounds like you have something called a five workflow framework that you can talk us through that will give our listeners some peace of mind as they think about deploying agents.

Nikunj Bajaj: 22:24 For sure. Actually, I talked about this. I've also written a litle bit about this five workflow framework and let me describe what this ... So this is basically a hot take that I have about how you build agents and set yourself up for success. And the hot take here is that most natural recommendation to getting started with something would be start small, like start building out some small things, see how it goes, you learn from that and keep building it towards something bigger. I feel like that start small approach hasn't worked very well with agents, especially in the context of large companies. I'll tell you why the start small framework leads an enterprise to come up with 500 different small workflows that can be automated using an agent, let's say. All of these different business units building out 50 agents each and now you have a spreadsheet of 500 agents and you're tracking which one is in production, which one is in dev, which one is in staging, whatever.

23:32 But each of these agents are low value enough that you cannot justify building a plumbing layer to think about

the entire governance aspect of your agent, or you do not learn enough critical information through this kind of small agents that you're able to build out a solid enough framework. On the other hand, I say, forget about those 500, think about a five workflow framework where these five workflows are critical to your enterprise, probably going to create dozens or hundreds of millions of dollars worth of impact or whatever your scale is. You can remove a few zeros or add a few zeros depending on what organization we are talking about, but think about the five critical workflows and pick one or two of them as the main one that you're going to automate using agents or almost automate. And when something is business critical, you will have enough bandwidth to think about building a framework or a governance layer or the plumbing layer on top of which you are able to now actually then build out the other 500 agents that I was requesting you to ignore in the first place.

24:41 So I think if you flip the order of do not start small, actually start big with that big, build a plumbing layer and then cover all the other small ones that sets you up for success. And we have seen this happen with a bunch of our enterprise customers where they started with some of these hundreds of millions of dollar worth of agenting impact. They ship that to production, they've been running it for a year and a half and now every little team is building out dozens of these agents, each of which are individually creating dozens of millions of dollar worth of business value impact.

Jon Krohn: 25:09 Did you see hundreds of millions?

Nikunj Bajaj: 25:11 That's right.

Jon Krohn: 25:12 Oh my goodness, that is wild.

Nikunj Bajaj: 25:14 Yes.

Jon Krohn:	25:16	I am not personally aware of any agents working on that scale. That is really cool.
Nikunj Bajaj:	25:22	Yeah. There have been times where we have gone to these dinners where founders and execs from very large companies have come in and even they are very surprised that something like this is happening. And I feel like the key insight to why this is happening is flipping this framework.
Jon Krohn:	25:40	Totally. And I admit myself, one of the key pieces of advice I give as the CEO of an AI software consulting firm is often start small, but I can see exactly why it makes sense to turn that on its head as well and say, let's get some big impact, otherwise it's not worth making the investment in some of this tooling. Yes. Really cool, Nakunj. For people who want to follow you after this episode, what's the best way to do that?
Nikunj Bajaj:	26:07	Yeah. Well, the best way to find us is on our website, truefoundry.com, T-R-U-E-F-O-U-N-D-R-Y. And you can also connect with me on LinkedIn.
Jon Krohn:	26:16	Fantastic. Yeah, we'll be sure to have your social links in the show notes and obviously your company website and you have listened to the podcast so you know that I always ask for a book recommendation at the end of the episode.
Nikunj Bajaj:	26:29	Absolutely. Well, I will actually give you two recommendations.
Jon Krohn:	26:33	All right, I'll allow it.
Nikunj Bajaj:	26:35	So one of these books have become now very popular, but I read it like years ago, Project Hail Mary.
Jon Krohn:	26:41	Have you read the book? It's like my next fiction book to read because I saw the film. I loved it. So many friends

have been talking about reading the actual book and how great it is.

Nikunj Bajaj: 26:51 Yeah. I waited for the book for years and I absolutely love reading science fiction. And the thing about Project Hail Mary that I find very interesting is there's this person who is trying to solve problem against all odds and just keep at it where the problem just seems impossible to solve. So that's one very interesting thing. I find it very relatable with the startup journey and well too similar end, like there's this book called The Hard Thing About Hard Things, which literally puts it in your face how hard it could be to build a startup and go through that. So I think that's the other one that I really like.

Jon Krohn: 27:27 I love that Nukunj. Thank you for those great recommendations, inspiring recommendations indeed. Thank you for taking the time out of your busy day. For me and for our listeners, I learned a lot and yeah, hopefully we can have you back on the show soon to hear more about how your skyrocketing journey is going.

Nikunj Bajaj: 27:43 I absolutely appreciate it. And thank you so much for having me here, Jon.

Jon Krohn: 27:46 Of course. Wow. I certainly enjoyed that conversation today with Nick Kunj Bajaj in it. He covered how his startup True Foundry is allowing clients like Nvidia and Siemens to realize returns on investment of over a hundred million dollars on Agentic deployments, including through their AI gateway and its three LLM, MCP and agent components, as well as his five workflow rule. I hope you enjoyed the conversation as much as I did to be sure not to miss any of our exciting upcoming episodes, including the upcoming episode, 1,000. Be sure to subscribe to this podcast. But most importantly, I just hope you'll keep on tuning in. I'm so grateful to have you listening. Until next time, keep on rocking it out there

and I'm looking forward to enjoying another round of the SuperDataScience Podcast with you very soon.