



SuperDataScience

SDS PODCAST

EPISODE 985:

**The Four Types of
Memory Every AI
Agent Needs, with
Richmond Alake**



- Jon Krohn: 00:00:00 In order to build an effective agentic AI application, there are lots of things you have to get right. But one area that particularly requires a lot of attention and has not been getting enough of it in recent months is agent memory. Welcome to another episode of the SuperDataScience Podcast. I'm your host, Jon Krohn. Today I'm joined for the third time on this podcast by Richmond Alake, and this is an episode dedicated to agent memory. So you're going to learn all about it, why it's so important, and solutions for nailing agent memory, including lots of free open source resources provided by Oracle, where Richmond works as an AI developer experience lead. And you'll also hear about courses that he's provided with Andrew Ng at deeplearning.ai, as well as in the O'Reilly platform. So you'll come out of this episode with everything you need to understand the basics of Agent Memory and how you can go further to build really powerful agentic AI applications.
- 00:01:04 Enjoy this fun conversation with Richmond. This episode of Super Data Science is made possible by Anthropic, Acceldata, Cisco, and Oracle.
- 00:01:16 Richmond, welcome back to the SuperDataScience Podcast. Last time we met up, we were live together in London about a year ago. People can check out episode 871 from March of last year. It's a great episode with you. Today, we're live in New York. What's happened over the past year?
- Richmond Alake: 00:01:34 Well, a lot has happened. Firstly, I'm at Oracle, Director of AI Developer Experience, which is a fancy title for just saying I focused on AI developers and AI engineers and getting the Oracle AI database over to them when you build in their AI workload. But in the space of AI in a year, a lot has happened.
- Jon Krohn: 00:01:54 For sure.



- Richmond Alake: 00:01:55 We're going to get into that.
- Jon Krohn: 00:01:56 Are you still based in London?
- Richmond Alake: 00:01:57 Yes. Outskirts of London, but easier to say London.
- Jon Krohn: 00:02:01 Nice. And what are you doing in New York this week?
- Richmond Alake: 00:02:03 Well, doing a lot. As in just this morning, we just had a meetup for AI memory engineers and developers working on agent memory. We just did a meetup of AI memory breakfast. A few folks turned up and the insights there was very good. I've changed some opinions on what I had around agent memory. We spoke about implementation detail. We spoke about high level stuff as well. Then this week, we have the Oracle AI World New York, which is a Oracle event where we have our customers, our developers coming into ... We've got the Javits Center. Sure.
- Jon Krohn: 00:02:40 That's the biggest place you can book in Manhattan for this kind of thing.
- Richmond Alake: 00:02:42 Exactly. We got that. Two days is going to be all Oracle and just really doing workshops, hearing what Oracle was talking about in terms of AI and agents, and we're going to get to meet our customers in person and developers in person. That's the real reason I'm here. But while I'm here, I usually take some time out to do some talks, talk to you as well, and we're going to be doing a bunch of meetups as well. I like how
- Jon Krohn: 00:03:02 This is your time out. It's coming and talking to me on a data science and AI podcast. And we're going to be talking about these new things you've been learning about agent memory, not just over the last year, but the stuff you've been learning in the last few days. For listeners who are new to the concept of agent memory, what is it and why should they care about it?



- Richmond Alake: 00:03:22 Well, I'll define it very quickly, then I'll talk about why they should care about it. Agent memory is this encapsulation of building a system that essentially learns and adapts of new information. And the term encapsulates systems that work together to allow for adaptation of AI agents. And the systems are your embedded models, your rerank car, your database, and your LLMs. Those are the systems that will hold data, move data, and store data. So that's what agent memory encapsulate and that's what it is. Why should developers care? Well, it turns out memory is one of the critical components of building AI agents that actually work in production and actually create value. And I always use a human analogy whenever I talk about memory. Memory is important to humans and in AI we are trying to replicate human intelligence, then you can connect the dots and figure out that memory is going to be a critical component of any computational systems we're building.
- 00:04:23 So developers should definitely be caring about memory.
- Jon Krohn: 00:04:26 A big trend we've had on the show in recent months is guests coming on to tell us how important it is that agents have the right context in order to actually be useful in a business. We know that LLMs are extremely powerful. Anybody today using frontier LLMs, even open source LLMs, really powerful, obviously. But if that agent doesn't have the right awareness, situational awareness of what's going on in the business and how it should be functioning in a particular circumstance, given the kind of expected behavior, given the history of what's been going on, that agent is going to misperform. And it could be a really disappointing node in some kind of expectations for the business. And it sounds like memory, figuring out what to store, how to be able to look over those memories to dig for the right context at the right time is the key.



- Richmond Alake: 00:05:24 Yes. Yes. Spot on. And in fact, very spot on in how you described the real job to be done, which is how to store, what to store, when to store, and how to retrieve and how to forget. That is the work we see a lot of developers doing and spending the most time on. And in the space today, there isn't a standard way of doing this. There isn't a standard way to model memory in an agentic system, which means that this is an exciting space to be and that a single developer could create a solution that can go up against the Frontier Labs. So that's why it's such an exciting space.
- Jon Krohn: 00:06:01 Yeah. And speaking of it being an exciting space, you find it so exciting that you have coined a new term for people who work in this space, you call them memory engineers. Tell us about how you coined that.
- Richmond Alake: 00:06:13 Yeah. And the term memory engineers just came out of necessity, which is to bring along all of the database engineers and information retrieval folks that have been working on getting the right information from repository of data into systems. They've been doing this for decades, but bring them along into this AI era of agents. And at the same time, take the folks that we have today, the AI engineers, and just expose to them the things that we've been working on for decades within a database community. So I find that with this new way of looking at data within the agentic context or within the agentic boundary of a system, which is memory, we can actually draw parallels to say, Hey, there is a engineering discipline that can be concerned with how we manipulate data within agentic system and we encapsulate that in agent memory.
- Jon Krohn: 00:07:06 For people who are listening, it seems like a great place to get started with moving in the direction of becoming a memory engineer. Not necessarily the job title, but just having some of the key skills that you've already highlighted and then we'll be discussing over this episode.



It seems like a place that they could easily get started is by going to your hashtag on LinkedIn, which is 100Days of Agent Memory. Tell us about that initiative. Is a hundred days over now?

Richmond Alake: 00:07:34

Yeah, it's over now. It was one of the hardest thing I've ever had to do on social media. I did it and I didn't do a hundred days consecutively. There were days that I missed, but I did it just to bring awareness to agent memory in general and let people just to educate the masters. And you find that that's what we do over one, the team over here at Oracle, we're always educating developers. And one of the things we're definitely educating developers on is agent memory, and that's what the hundred days was. And interestingly, it was also for myself selfishly, because I wanted to really dedicate myself to exploring this space and do it out there. Learning in public is always good, but it was also for myself. And you will see that there are some insights and some of the insights I shared maybe in the earlier days probably changed later on and it's always good.

00:08:25

And this is the best thing about the series. It's long enough for there to be certain key events in the AI space that validate the trajectory that we're on. So in the space of the hundred days, OpenAI and Anthropic did stuff around memory, which validated the direction we're taking things on. Folks started bringing out memory solutions from research all the way to application. We saw startups raising millions, which shows that there is a bit of a tension there in terms of investment. But so over that hundred days or hundred and so days, it was really good to go through that journey and people came along with me, which is always good as well.

Jon Krohn: 00:09:04

Perfect. Yeah. And you're preaching to the choir to me when you talk about how important it is to build in public and the value of that. A lot of people are probably aware of me today as the host of this podcast or a book writer or



video content creator, that kind of thing. But the whole point, the only reason why I was doing that in the beginning was to force myself to learn this kind of commitment to say a hundred days of agent memory and you commit to that and you force yourself to get that out in public. Otherwise, you feel embarrassed to not follow through on this public commitment that you made. Otherwise, you can end up in situations where if you just make this commitment to yourself, you're like, for a hundred days, I'm going to study agent memory. It's very tough to stick to that if you're just doing it in a notebook, doing it on your computer because it's so easy to say, "Ah, well, I'll skip today." And then all of a sudden a week goes by, a month goes by and you only got 10

- Richmond Alake: 00:09:55 Days into your
- Jon Krohn: 00:09:56 Hundred days. But when you've got the hashtag, when you've got the presence that you have, Richmond as well, you got to keep going. You mentioned that it was one of the hardest things you've done in terms of content creation. Why was it
- Richmond Alake: 00:10:06 Challenging? Consistency, right? So you've never done a
- Jon Krohn: 00:10:08 Hundred days kind of challenge before?
- Richmond Alake: 00:10:10 I've never done a hundred days of anything in public. And the closest I've got into is just going to the gym consistently for a hundred days, which I'm failing at now to be fair. But really it was also challenging because I have my day-to-day job, which can get really, really ... Which also has to do a lot with memory. So it's not that I wasn't thinking about memory every day. Every day, I've worked on memory for maybe every day for the last two years. I can say that. And it was more just writing in public and just going through that mental barrier just to put some content together that we actually do well. And the content were not just words. They were cold as well



because over in Oracle, we actually practice what we preach. We focus on the Oracle AI database and we see databases that unified Agent Memory Call and we experiment and we educate our developers and give customers different solution.

00:11:09 We actually have a technical resource on GitHub called the Oracle AI Developer Hub. And if you go there, you're going to see loads of resources from the team, from myself, just we're at the front lines of this space. So yeah, that's why it was difficult. It was difficult because writing and thinking is creative, but we were always doing.

Jon Krohn: 00:11:30 Right, right, right. And I suspect you might've had other commitments, of course, as well. You're doing things like traveling to New York and hosting events and you're like, "I need to find time in my hotel room to get today's post out. " Yeah, I can see why that would be challenging for sure. Well, congrats on finishing it up. All right, let's talk more about technical details around agent memory. So I think most people, probably most of our listeners, think of AI agents as stateless. Why is memory the missing piece and what breaks when agents don't have memory?

Richmond Alake: 00:12:02 Okay. So the best way to explain it, and it's a terrible example, but human me here for a second, right? I use the example of a travel agent, like a human one, not agent agent, a human agent. I don't know if anyone still uses travel agents today to book holidays, but once upon a time there were people that would, you go into this place and you would go there and they would get your holiday package. So imagine yourself, John, you wanted to go to a holiday and have you been to Tokyo?

Jon Krohn: 00:12:31 One time, and it was great. I would love to go back. If you

Richmond Alake: 00:12:33 Were



- Jon Krohn: 00:12:33 Thinking of arranging a trip for me, feel free to go ahead, Richard.
- Richmond Alake: 00:12:37 Exactly. I think we have one of the Oracle AI awards in Tokyo maybe.
- Jon Krohn: 00:12:40 Nice.
- Richmond Alake: 00:12:42 So imagine you go to this travel agent and you say, "Hey, I'm Jon. I like places that's culture. I like to eat a lot. I like tradition and I like Tokyo essentially. And I want you to remember my preference and go get me a holiday package for the Tokyo destination." And you do this on Monday, right? And you go back, you come back on Wednesday and you're expecting some holiday packages, but you get nothing. In fact, the travel agent says to you, "Who are you? " What would you do? This is a rubbish travel agent. Well, the same thing can be applied to your actual computational entities, the agents that we actually build. You want to interact with this agent. And if you go away and you come back and you have to start from scratch, it is not a good experience. And a lot of software we're building today are very customer facing.
- 00:13:37 They have chat interfaces and they're meant to be interactive. But if you have loss of continuity in between sessions, you lose your customer, you lose the value that you're trying to give. And that's why memory is important. So that's the best way I can describe it where I bring along non-technical and technical bits, but we can-
- Jon Krohn: 00:13:58 Right, right. Well, yeah, let's double click exactly onto the technical bits. This next question is one of my favorites that we have planned for you throughout the episode because it's something that ... I have a neuroscience background and so these are the kinds of things doing a neuroscience undergrad in particular where you're just kind of serving across different psychological disciplines. I spent a lot of time learning about different kinds of



memory. And so it's interesting now to see a lot of those same kinds of animal memory systems being replicated in agents. And so according to you, agent memory can be broken down into four categories, episodic, semantic, procedural, and working memory. Can you break down each of those for us here?

- Richmond Alake: 00:14:42 Yeah, yeah. So first thing I wanted to point out is this is the first time I have someone that can fact check me on all of this, right? So this is good.
- Jon Krohn: 00:14:52 Live on air.
- Richmond Alake: 00:14:53 Exactly. Because one of the things that we've done as, I guess a society or humankind is look to nature to inspire our technological advances. So I always use the example of planes, right? The Wright Brothers had inspirations from the early inventors of vehicles or flights that actually took inspiration from birds. So you always look to nature to try to inspire technology. Same with convolutional neural networks. There was an experiment by Aberlong Weiser back in the 1970s where they experimented with the visual cortex of cats to realize how neurons work. But anyway, the long story is we always draw from biology to inspire our technology advancement.
- Jon Krohn: 00:15:42 Yeah. If you don't mind me interjecting
- Richmond Alake: 00:15:43 Just for a
- Jon Krohn: 00:15:43 Moment. Yeah. That Hubble and Visel experiment absolutely a critical moment in the development
- Richmond Alake: 00:15:49 Of- Jon, I'm so excited. This is good because I always talk about the experiment. Not everyone knows it, but of course you know it because you did neuroscience and you did. I did, but it
- Jon Krohn: 00:15:57 Was also, this was the Hubble and Visal experiments. They were a key ... And for years when I was ... About a



decade ago, when I started teaching deep learning to the public, when I was giving talks to technical or non-technical audiences, I would often use the Hubel and Wiesel experiments to illustrate how neural networks work when you layer deeply because yeah, they had ... It's a really interesting story. If you don't mind me taking a couple minutes

Richmond Alake: 00:16:25 To explain

Jon Krohn: 00:16:25 This. It's all easy.

Richmond Alake: 00:16:27 We'll switch rolls.

Jon Krohn: 00:16:30 Hubel and Wiesel, they were trying to figure out how the vision system worked and they could see in the cat's brain that there were all of these nerve fibers going from the eyes to the back of their brain, the visual cortex. And so they tried for ages and ages to ... They would have a cat sitting in a harness, they'd have recording electrodes in

Richmond Alake: 00:16:52 What

Jon Krohn: 00:16:52 They could see was the right part of the brain, and they would show different kinds of images to the cats. And day after day after day, there'd be no readings until many of the great discoveries and histories like X-rays, they discovered by accident, they were frustrated at the end of the day. And instead of taking the cat out of the harness before starting to take apart the rest of the recording equipment, they removed a slide that they'd been showing on the projector and the slide had a straight line edge. And as that straight line edge passed through the cat's field of vision, boom, the neurons lit up. And that's how they made the discovery that the first brain cells that receive information from the eyes, they're responsible only for straight lines at a specific orientation. And then they later discovered that that information, what they called



simple cells, detecting those straight lines, it gets combined into edges and corners to get a second layer, and then that second layer gets combined into a third layer to get even more complexity, more abstraction, which is exactly how deep learning systems work.

00:17:59 So yeah, convolutional neural networks, exactly.

Richmond Alake: 00:18:01 The early layers there, they capture the edges, the features of the edges, then as you get through the deeper layers, you see the abstract shape starts to form. I'm so glad that you explained that experiment better than I have ever done. Well,

Jon Krohn: 00:18:15 It's kind of wild where we've come so far with agents and this kind of abstraction where like a decade ago, it seemed so important for us to understand how neural networks and deep learning work. And now you don't even hear people talking about deep learning that much, even though that's how LLMs work, which is how agents work. It's kind of interesting how you just keep tacking on more and more to get more complexity out of these systems, more abstraction.

Richmond Alake: 00:18:36 Yeah. And going back to the whole form, the whole agent memory types that we mentioned, right? Semantic, procedural, working memory, and what was the other

Jon Krohn: 00:18:48 One? Episodic, semantic, procedural, and working.

Richmond Alake: 00:18:51 Yes. Okay. Your

Jon Krohn: 00:18:51 List. I'm just reading

Richmond Alake: 00:18:52 It. Exactly. My list, right? I'm going to go, my memory's not that great then. I'm going to go through them and just talk about examples of how they could manifest or how they could be implemented. So let's go and keep me on a stair with the list. Let's go. It was a long flight from London to New York, so I'm a bit slow. And



- Jon Krohn: 00:19:11 You're working without notes. For people who aren't watching the video version, Richmond is just freewheeling everything
- Richmond Alake: 00:19:16 Here. Well, the hundred days of agent memory helped. All right. So let's go with episodic. Episodic is one of the easiest ones to actually understand. Episode are essentially memory that have some association with time. And the easiest one within an agent is back and forth conversation. So the actual conversations you're having for developers or engineers listening, this would be the user would be the assistant or the assistant or the user, then the concept will be the message, then you have a timestamp. The timestamp is always important in episode because that's how you recall that particular type of memory type. Then that's episodic. Another example would be procedural. One thing that's very popular today is skills.mdfiles.
- Jon Krohn: 00:20:01 Let me just quickly, before you move on from episodic, maybe I'll try to give an example. So you gave the travel agent example. So the instance of somebody, if you imagine a human travel agent and you're a human travel agent, somebody comes into your office on Monday, they say, they describe the kinds of things that they're looking for, cultural, lots of great food, that kind of thing, set up a trip for me and they say, "I'll be back on Wednesday and hope you have something ready for me. "It's an episode. It's a specific ... If you were the travel agent, you could mark that down in a notebook on Monday at 10:00 AM Richmond came into my office. It's a discreet episode where you can, as a human, you can think back by Wednesday, you should be able to think back to Wednesday or think back to Monday and think, " Yeah, at 10:00 AM, Richmond was here in my office.
- 00:20:52 I remember what he's looking for. "So you recall that episode.



- Richmond Alake: 00:20:54 Exactly. Nice. So let's go over to procedural memory. An example of that would be within the agentic context would be maybe workflows. And I have another content, another example of procedural memory called using the database as a toolbox. But let's talk about workflows. You could think of things like skills.mdfile. They're very popular now at the time of this recording. And skills.mdfile are basically marked down that covers instructions on how to do certain tasks that are given to an agent. I call it the SLPs for agents, essentially.
- Jon Krohn: 00:21:30 Yeah, standard operating procedures.
- Richmond Alake: 00:21:32 Exactly. But these are basically procedural memory because humans have the same concept of this routines and skills. We store them in a particular part of our brain, and you're the neuroscientist there, so you can fact check me on this. We have this part of our brain called a cerebrum, cerebellum.
- Jon Krohn: 00:21:49 Cerebellum.
- Richmond Alake: 00:21:50 Cerebellum. Yeah. And that stores all our routines and skills. So you can actually think about that in the same context of agents as well. We are writing a bunch of skills as MD file and we're storing them in people are using file systems and we could talk about file system versus databases for agent memory. But over in Oracle or over in Oracle, we're experimenting with putting these skills into the database and actually putting them within tables that have some different representation of this data, one of them would be a vector representation, and then we can progressively expose their skills at the right time. So rather than giving the agent a hundred skills or MD file, within a context, we can just retrieve the actual skills that we need at the time, which allows you to start to scale essentially. But that's an example of procedural.
- 00:22:40 Do you want to do it? Yeah, I'll



- Jon Krohn: 00:22:41 Try to come up with an example here. So if you think of some kind of thing that you do regularly, like making scrambled eggs,
- 00:22:48 It's not that complicated and you probably don't remember the first time somebody showed you how to scramble eggs. It's not like an episode that you're like, oh yeah. I mean, maybe there might be some people who are like, "I love the time and grandma showed me how to scramble eggs." You might have the kind of episode that you can recall, but for most people, it would just be kind of something you can make scrambled eggs now and you don't really remember how that happened, but you've just kind of figured out through experimentation, maybe some things you read online or some things people taught you, a cookbook. There could be all these different inputs that over time created this procedure that you're aware of. There's not any specific episodes tied to that procedure. You just know how to scramble eggs.
- Richmond Alake: 00:23:28 Exactly. Exactly. That's a very good example. So let's go to semantic. Semantic can be a bit easier. So semantic is actually just, the easiest one would be world knowledge. You can have knowledge about a certain topic, that's semantic memory. And for agents, that could be all of the, let's say the institutional knowledge within your enterprise data, how to do certain things, some abbreviations or some words. And you could just bring that into your agents to get them to actually start to have the knowledge that the same your employees have in terms of that particular use case. So that's semantic memory. There are different types, but that's the one we're going to use today. So over to you.
- Jon Krohn: 00:24:07 Yeah, exactly. So this is kind of like any kind of knowledge that's in an encyclopedia, like
- Richmond Alake: 00:24:13 Wikipedia,



- Jon Krohn: 00:24:14 These kinds of key pieces of information. Again, so it's different from procedural in that it's not necessarily a sequence, it's a fact, it's an association.
- Richmond Alake: 00:24:24 Exactly.
- Jon Krohn: 00:24:26 Richmond lives in London. It's like, I know that's semantic fact. It's not associated with a specific episode. It's not associated with a procedure. It's just a piece of information that I know.
- Richmond Alake: 00:24:36 Exactly. And then we can talk about working memory, which is another type of memory. Working memory is this short term, specifically a subset of short term memory. Working memory is what you're using in real time in the context. There's a bunch of information that is my working memory that I'm using to speak to you right now to actually interact with you. I'm not having to think for an extensive period of time. So that's my working memory. The best way I would describe this within agentic context is the context window of the LLM. That's working memory. And over to you.
- Jon Krohn: 00:25:09 Yeah, no, that makes perfect sense in the context of an LLM. When you think about it from a computing perspective, you're not going to need to do some kind of retrieval. You're not going to need to look over a database or do some kind of rag process in order to find whatever's in the working memory. It's in context in the agent. And this is analogous to the same experience that we have when you stay focused on some complex task. Right now, I'm trying to find as much time as I can in a given week to be writing my next book, which has been a difficult thing to find time for. But it's really important that I carve out time to be focused on book writing for several hours at a time because it can take me up to an hour to get all of the relevant context back into my working memory in order to make progress for hour two, hour three on the book



- Richmond Alake: 00:26:02 Because
- Jon Krohn: 00:26:03 There's so many. Where are we in the book? What do readers already know? Where are we going next to get all that kind of stuff just in my working memory? It takes a bunch of time. And yeah, it's also for agents, it ends up being one of the key pieces of memory to have in place.
- Richmond Alake: 00:26:19 Yes. Yes. This is good. I like the back and forth. Very educational. But one thing your listeners can actually get is we cover all of this and more in a recent course we just did in partnership with Deep Learning AI. So Oracle and Deep Learning AI along with Andrew Ng, we just released a course on building memory aware agents.
- Jon Krohn: 00:26:39 Yeah,
- Richmond Alake: 00:26:40 It's so
- Jon Krohn: 00:26:40 Cool. I saw on LinkedIn when you guys released that course, you had the announcement, you and Andrew and together, that's pretty cool.
- Richmond Alake: 00:26:46 Yeah. And a couple of my teammates as well, Nacho. And this is something that is very interesting about Oracle. And I joined Oracle, it's been about five to six months now, is there are so many people that understand the agent memory application and how useful they can be and actually build an AI application. And when you look at Oracle, Oracle has existed for over four decades and it started off as one of the oldest database companies. So I should not be surprised that they understand how things are shifting and the paradigm of change itself. So we were very quick to actually take on this agent memory narrative and start to meet developers where they are. So the course we did with Andrew answers a lot of the questions we saw developers asking and our customers asking and what we think are the patterns that will begin



to solidify themselves to what might look like a standard, which I'll obviously like to contribute to.

Jon Krohn: 00:27:41 Of course.

Richmond Alake: 00:27:42 But also we recently announced that we're going to be putting out agent memory SDK in Python for developers that like to get a bit hands-on. And we also have a private agent factory that allow people to actually build agents using no-code tools. So Oracle has a massive understanding of the shift that is happening and how to actually bring these to developers and our customers.

Jon Krohn: 00:28:06 Fantastic. And we'll make sure to include links to all of

Richmond Alake: 00:28:09 The

Jon Krohn: 00:28:09 Items that you just mentioned in the show notes for our audience members to be able to enjoy. It sounds like you get to have a lot of fun working with clever people at Oracle creating useful materials for our listeners.

Richmond Alake: 00:28:22 Yeah.

Jon Krohn: 00:28:23 To dig into a buzzword that I said not too long ago, I mentioned RAG and you actually, you'd noted here how agent memory goes beyond retrieval augmented generation. Where do you see RAG falling short for agents?

Richmond Alake: 00:28:41 Well, RAG is retrieval essentially. It's just retrieving information and passing that alongside with the user prompt or user objective to actually augment the generation of your LLMs. It's a technique that's very useful to actually ground the LLM in domain specific data. Memory, you do need to retrieve memory, but you also need to update memory, you need to consolidate memory, you need to resolve conflicts between memory. You actually need to forget memory. RAG doesn't cover all of that, right? So- I think



- Jon Krohn: 00:29:13 A good example of where it could break down is my previous startup that I worked that I was a co-founder at until about 18 months ago, we were specifically building applications in the human resources space. And you could imagine it's kind of easy to think, okay, if I want to have this human resources agent working, let's just throw all of the HR manuals that we have in the business into the memory and do rag over that. But the conflicts are a big issue because there can be directly conflicting information
- 00:29:44 In these HR manuals. One HR manager creates one in 2023 and then another one creates one in 24, and how you're supposed to resolve some specific kind of HR incident, there could be directly opposing information. And so that is one of the kinds of situations Or where I can fall short because you can retrieve one piece of information, the 2023 information and tell a person to do it one way on the first day. Exactly. And then second day come back and tell them to do it another way. In both cases, they were retrieving as far as they know correct information, but conflicting.
- Richmond Alake: 00:30:16 Exactly. RAG is a useful part within the toolbox of agent memory, but there is so many more, which again, we cover in the course, we cover in our technical assets, we cover in different blogs that we put out as well on our Oracle developer resource.
- Jon Krohn: 00:30:33 Nice. Is part of it to have in the working memory some kind of consolidation that resolves these ambiguities and these complex-
- Richmond Alake: 00:30:41 Okay. We can get into the technicalities. I wouldn't have that in a working memory because actually resolving some of the technicalities and conflicts can be computationally expensive. And this is why- And you probably



- Jon Krohn: 00:30:53 Want it to be permanent.
- Richmond Alake: 00:30:54 Yeah. When you want to store it, you want it to be permanent. You also want to think about latency as well. You don't want your user waiting for your system to figure out which memory to use and how to resolve conflicts, then give it ... That's a bad experience. This is why memory engineering is such a unique, not unique, but it's an important discipline. But again, nothing I've said is new. Latency and the reduction of latency and retrieval pipelines or information system, we've been trying to solve that for decades within the database world. So there are techniques like quantization. If you're using vector embeddence, you can reduce the precision data types of your vector embeddence to speed up the retrieval or reduce the storage space utilization. So these are things that might get lost in people that are just coming into the space. If you're vibe coding an AI application today, you are not going to know about quantization.
- 00:31:50 So you really need to actually start to understand these and start to educate the new people coming into this space, but as well as bring the people that have been here for decades along with us.
- Jon Krohn: 00:32:01 For sure. I think in this vibe coding paradigm, it is still valuable for the human vibe coder to have some understanding of how these systems work and how they could be improved. Maybe it will even be important for people to still understand the Hooble and Visal stuff
- Richmond Alake: 00:32:16 That
- Jon Krohn: 00:32:16 We were talking about earlier in this episode. And in terms of being able to understand what's possible, because yeah, you can end up, I think probably for years to come, I could end up being wrong, it's such a fast moving space, but there's a broad context and an understanding of where a product or an experience could



go to that the LLM's not necessarily going to be able to come to a conclusion on its own. So yeah,

Richmond Alake: 00:32:46 Get out

Jon Krohn: 00:32:47 There and learn from Richmond's courses.

Richmond Alake: 00:32:48 Yeah. And one of the things that actually dawned on me when I was speaking to the developers and startup folks working on Agent Memory in New York this morning was they shared with me that they don't see a general purpose solution to memory because memory is so unique to different use cases and different niches and different workflows that they think the ideal solution will be different ways of handling memory will probably emerge. And I don't think there'll be one general solution. Which for me, standing from an infrastructure point within the stack, I'm thinking more of the general application of the solutions that I'm creating. I want to solve memory for all. And today I was hearing, well, it might be a bit harder. How about we solve memory for this specific use case and then we just keep tackling things on. And that was a very interesting insight and it kind of shaped how I think I'm going to be looking at solutions and educating developers.

00:33:52 And you'll probably see more of these in the next coming week for the listeners in the events that we're going to be in. So we're going to be ... One of the biggest events, which I'm sure a lot of AI folks go to is AI engineers in San Francisco. So we're going to be there. It's June and July this year, so we're going to be there. So if you see any Oracle folks, just say the two words, agent memory to them and they'll go off.

Jon Krohn: 00:34:17 Nice. I like that. And it sounds like you also have, we're expecting this episode to be out April 21st. It sound like you have an event next week, a time of release.



- Richmond Alake: 00:34:26 Yeah, so we do. The one thing that we are trying to do over here in Oracle AI database is meet developers where they are. So we are meeting them in terms of where they're actually educated, where they're getting information from, which is why we partnered with Deep Learning AI. We're in some of these critical events as well, AI engineers. And one of those events is AI Developer Conference in San Francisco, also by Andrew Ingen de Planning AI. So our aim is to just sell developers and meet them where they are.
- Jon Krohn: 00:34:57 Fantastic. Yeah, lots of great events for listeners to check out in person. Kind of back to the technical stuff
- Richmond Alake: 00:35:04 Around
- Jon Krohn: 00:35:05 Agents, let's broaden out a little bit from just agent memory to the broader stack where memory is a key component of that. There's a lot of noise around agent frameworks, orchestration layers, tooling. How do you, Richmond, think about the agent stack today and beyond memory, what are the layers that really matter?
- Richmond Alake: 00:35:27 So the layers in the agent stack, right? So one thing is very clear is the agent stack is expanding, but also at the same time, it's consolidating. But the layers that really matter, we have the orchestration layer, we have the connectivity layer, we have security, right? That's definitely an important bit. We have the reasoning layer, we have the memory core, which is what will be where you're storing your data. We also have the memory managers as well within this agent stack. And I'm sure I'm missing some layers, but the key thing I'm seeing with all of this orchestration, reasoning, memory managers, the way the field is now defining this whole space or how they're actually implementing the agent stack is used, they're using the word agent harness to describe all of bringing all of this layers and tool sets within this layer together into a solution.



00:36:16 That is

Jon Krohn: 00:36:16 Something really recent, isn't it, that everyone seems to be calling it an

Richmond Alake: 00:36:19 Agent

Jon Krohn: 00:36:20 Harness. Yeah.

Richmond Alake: 00:36:20 It's recent and it's popular because people were saying it maybe about a year ago, we're in April now, right? Yeah.

Jon Krohn: 00:36:30 Well,

Richmond Alake: 00:36:31 Yeah. Memory. Exactly. Exactly. We're in April now. So I think Anthropic put out a piece last year in May on effective agent harness or effective harness. People have been talking about building systems that work together to create this outcome of maybe some form of agent execution that you want. We all focus on coding agents. So if you use CloudCode or use Codex, those are agent harness around the specific models to actually allow for a specific outcome. So this is what the field is defining all of the key parts of the stacks working together. And I think understanding your agent harness is important, understanding how you can customize your agent harness is important. And obviously, if you're talking to me, I'll tell you to go build a memory first agent harness where you're thinking, where you're approaching the implementation of your agentic system by just thinking the information that comes into this system needs to be recalled and forgotten.

00:37:36 If you just take that mindset shift, it just changes the way you approach every, how you model your data, the selection of tools as well. For example, you start to think, okay, if I'm going to be storing information, I probably don't want to be using multiple databases. You probably want to be using a database that can handle all of the different heterogeneous nature of data that you're going



to see in production. You can use multiple databases. I've seen that anti-pattern happen in some of the AI teams I talked to, but I explain it as this. Our job is to reduce the cognitive load for LLMs. And when I say our job, I mean the developers, right? We're meant to give them all of this context so that the LLMs could focus more on the reasoning that they have to do. Well, if you take that out of the context window, if you add more infrastructure, more components to your infrastructure and complicate it, you increase the cognitive load for not just the LLM because now the LLM has to think about or have an awareness of multiple different databases, you're also doing it for your developers.

00:38:42 Because now your developers have to know how to work with multiple databases and all the code that glue them together. So reduction of cognitive load is this thing that I'm really trying to emphasize because in AI, we find that speed of experimentation, of implementation, it's a mult. You have to be in the arena, you have to experiment. And if you're thinking about how you're going to glue database one with the next database and what database should I use or query languages, then you're wasting time. The AI train is moving, it's moving fast.

Jon Krohn: 00:39:19 So what do you think, for our listeners, what do you think are the parts that they should be building themselves in relation to agents versus what they should be taking off the shelf?

Richmond Alake: 00:39:29 Yeah. So the one thing that is very important is understanding the domain that you're trying to build for. Workflow expertise is so important. It's so crucial. There is no way I can build a healthcare agent better than an AI engineer that works at maybe one of the large healthcare providers, because they have that domain expertise of everything that goes in there. I can give them some building blocks to help them think about it, but there's no way we can compete. Same thing with legal. So one



thing that the engineers today should be focused on is understanding the workflow you're trying to automate or you're trying to optimize, understand the workflow in detail. Then the one thing that you probably shouldn't be wasting, not wasting time on because two selections is important. And I do think looking for a general purpose database, one like Oracle AI database that can handle all of your different data types.

00:40:30 So using the Oracle database, you can handle vector data, you can handle graph data because everyone likes to do GraphRAG or knowledge graphs now. Especially

Jon Krohn: 00:40:37 For semantic knowledge.

Richmond Alake: 00:40:39 Exactly. Especially for semantic knowledge. And you want a database that can handle all of the relational data as well that you normally have and lexical and spatial, all of these different data types. So we can handle all of that in the Oracle AI database and we bring in a lot of what developers have to think about building this AI system. So within the Oracle AI database, you can have embedded models in the database. So that reduces some of the cognitive load on your developers to start thinking about, okay, how do I start building this ingestion pipeline with the right embedded models? We are looking to take away all of this concern so that our customers, our developers can really play to the advantage of speed of innovation.

Jon Krohn: 00:41:23 Nice. Thanks for breaking that down for us. Of all of the types of agent capabilities that are being incorporated into these agent harnesses, so things like planning, tool use, orchestration, do you think that memory has been underinvested in so far?

Richmond Alake: 00:41:38 I think there is a focus on top-in benchmarks from the research labs, right? They're very benchmark focused. There is also a focus on the particular coding agent use case. I do think memory was ignored initially because,



and I say this because two years ago when I was seeing definitions of agents, memory was always missing. There was tools, there was reasoning, there was planning, but no one mentioned memory. And I was like, wait, that's the most obvious component of an AI agent. I think memory came a bit later, but now everyone is paying attention. Everyone's realizing that we need to solve the ability for this models to learn. So there's agent memory, there's another paradigm. And again, nothing I'm saying is new. There is the whole school of continuous learning that's existed probably in parallel with AI essentially. So nothing is new. It's just bringing what's old and just surfacing it up so we can bring the right attention to it so we can solve it.

- Jon Krohn: 00:42:45 Yeah. I think some people were grouping memory in as another tool,
- Richmond Alake: 00:42:50 But
- Jon Krohn: 00:42:50 I think that's true. It doesn't give memory the importance that it deserves in so many Agentic applications.
- Richmond Alake: 00:42:56 Yeah. And I'm biased. I call it the final battleground, right? Because I think if you solve memory in a very meaningful way, just hearing what I heard this morning, speaking to the developers, they're just waiting for a solution that just handles the whole retrieval of information and forgetting of information and doing it very, very well in a way that the system is just reliable. Because you don't have to think about the way you remember things as a human, you don't have to think about it. So developers are looking for that. So I do think they're expecting it to come from the research labs in a sense, but they're also expecting it to come from the database company. This is why I'm talking to them. But I think it can come from anywhere in the agent stack and agent memory is one of the fields where you can have a



single developer create a solution that could probably compete with one that comes from a research app.

- Jon Krohn: 00:43:54 Sure. Maybe because it's so easy today, that's why we're seeing so much fragmentation. Do you think that the agent stack fragmentation is here to stay or do you think in the future we might consolidate in the same way that the web did?
- Richmond Alake: 00:44:10 Yeah. So is the agents that going to consolidate? You mean by just in terms of tool selection, in terms of players?
- Jon Krohn: 00:44:19 Yeah. So historically for the web stack, there were lots of different ways of doing things, but more or less today, people do it all the same way.
- Richmond Alake: 00:44:31 Yeah. Back when I used to be a web developer, the biggest battle was, are you going to use Angular or React? That's what kept people off at night. Then VJS came and wherever. But I do think that developers like to experiment with new tool sets that come out, but I do think that there will be some staple in your stack that you're just going to use whenever you want to use your models. As in model, the key model players now in the stack are, there are quite few that have loads, except if you go into the open source direction. I do think there's going to be some consolidation and not because ... And that's going to be because the tool selection is not where the fun is. All the value is what you're actually building getting to your consumers. And at some point we're going to have key players like Oracle that are very in the problem that we've cut for ourselves, which is agent memory.
- 00:45:27 We're putting a lot of resources to solve that, to solving that. And then when we do, using our solutions in wherever form factors they come in will get you to where



you need to be. So you don't need to essentially shop around.

- Jon Krohn: 00:45:40 Yeah. Yeah. So in the same way that Python became a de facto standard for doing AI backend programming, some aspects of the agent stack will emerge as de facto standards.
- Richmond Alake: 00:45:54 You somehow just pissed off a bunch of Java developers.
- Jon Krohn: 00:45:58 Do you think there's Java ... Well, I guess-
- Richmond Alake: 00:46:00 Yeah, it's Spring AI and existing.
- Jon Krohn: 00:46:04 I don't know how many listeners we have out there that are developing their AI applications in Java, but ...
- 00:46:12 I'm sure there are great advantages to going down that route. Someone will have to come on. Reach out to me if you are in that situation. We'll have you on to explain why. Nice. All right. So we've talked a lot about agents. Let's talk about the specific solutions that you're representing here. So you lead AI developer experience at Oracle, specifically around the Oracle AI database. You've mentioned already earlier in this episode, some of the key reasons why you think the Oracle AI database is a great solution for people building agents, things like being able to handle RAG and graph databases all in one place. What are the other reasons why the Oracle AI database is a compelling platform?
- Richmond Alake: 00:46:58 So one thing, and this is important actually, and I'm speaking as an AI developer here. You've probably noticed AI developers don't care about security as much as in-
- Jon Krohn: 00:47:11 It is a personal problem I have. That's kind of an afterthought for me, which is silly.
- Richmond Alake: 00:47:16 Well, it's because we have to experiment. We always have to move fast. We have to experiment. Look at the uptake



of OpenClow or even MCPs. One of the key advantages of Oracle AI databases, all of that concerns is taken care of. The Oracle databases existed for decades. We've had to really understand the nature of security and privacy in different regions with different regulation, with different compliance, and we've been meeting them for decades. So again, AI developers don't care. They're like, okay, yeah. Okay, how does this help me build AGI? How does it help me make my agent be more autonomous? The key thing that I tell developers is security is important. Privacy is important. When you start to deploy most of the systems in some industries that are regulated, like the finance industry, the healthcare industries, the compliance you have to meet that every single component of your AI infrastructure has to meet.

00:48:15 So using one that actually just takes care of it, allows you to do what I said is important, which is move to market much quicker. So the Oracle AI database, we take security and privacy as a top priority for us, and we remove that cognitive load again from our developers.

Jon Krohn: 00:48:33 Cool to hear about the Oracle AI database. With your role as leading AI developer experience at Oracle, can you tell us a bit more about what Oracle is doing to allow people who are building to benefit from what Oracle's doing?

Richmond Alake: 00:48:50 Yeah. So Oracle is actually, like I said, meeting developers where they are, meeting builders where they are. And one thing that we're doing is putting out a bunch of products and tools and features that actually allow developers to build quickly and at the speed of innovation. So one of the things we released quite recently, and I think I've mentioned it already, is the Oracle autonomous vector database. And this is a vector database solution that Oracle is putting out there, currently in private preview, and developers can use it for free today. You have a solution where you can store your vector data for free. But also, Oracle is actually in the space of actually



providing some tools, some open source tools that some of our developers can use for building agentic solutions. So one of them is agent spec, which is this solution to actually making agents portable across different frameworks.

00:49:48 Because one thing we saw was this sprawl of people building their agents using different frameworks like Queer AI or using LangChain or using LAMA Index or ASTAC. And we thought to ourself, we need to standardize this and consolidate it because in enterprise, we actually have to have a common way of talking between large organizations. And the team put together a open source tool called Agent Spec, and behind that is another framework called Wayflow. So we are definitely trying to meet developers where they are either by giving them some of the free resources that we provide over Oracle, or even just going in there and putting all the learnings we're showing in open source tools.

Jon Krohn: 00:50:32 That makes a huge amount of sense. It sounds like a great reason to go with the Oracle AI database. Obviously, we've talked about the deeplearning.ai course

Richmond Alake: 00:50:42 That

Jon Krohn: 00:50:42 You co-instructed on agent memory. You also run live trainings at O'Reilly. I think I might've mentioned that earlier in the episode, but I can't remember for sure. So people who have O'Reilly subscriptions through their work or school or personally, you can look out for Richmond's upcoming trainings.

Richmond Alake: 00:50:56 Let me actually talk about that.

Jon Krohn: 00:50:57 Yeah.

Richmond Alake: 00:50:58 So I started that series with O'Reilly



00:51:02 As just the main mission, which is educate developers about agent memory. And I started this around this time last year. Not a lot of people were talking about agent memory, and I thought it was very important. I thought, okay, let me actually start to educate people on this. And O'Reilly, they said it was a good idea. They've seen some early signals. We started off with two hours and we got a very good turnout. So we did about four iterations of that and increased the time because there were just more, more. And now it's turned into a two-day bootcamp with O'Reilly. What started off as two hours in a year is now a two-day bootcamp with five hours on each side talking about agent memory.

Jon Krohn: 00:51:43 Oh my goodness.

Richmond Alake: 00:51:44 And the next session is going to be in June 16 and 17. So the developers that we're seeing in the future are going to be very important because they're going to start to, for lack of better words, they're going to be the architects of intelligence. There's a book called The Architects of Intelligence that people could go check it out. That book is actually quite interesting. I'm going off tangent here. That book contains a lot of predictions of things that are going to happen in AI that I think 95% went wrong from really credible people. It's good architects of intelligence by architects of intelligence by Martin Ford, I think. It's a good book.

Jon Krohn: 00:52:21 And you're saying the predictions were 95% correct or 95%?

Richmond Alake: 00:52:24 No, wrong. Incorrect. Incorrect. But you still think

Jon Krohn: 00:52:26 It's a good book.

Richmond Alake: 00:52:26 I think it's a good book to see how the ... We had people like Andrew there, we had folks from Google interviewed. It was like an interview book. I think Demis is also



interviewed for that book as well. It's a good book because you get to see the minds of the greats within AI really flesh out how they're trying to predict what happens, but we're looking at it now and we see, oh, even this people that are in the arena don't even know, but it's a good bit because you get to one, give yourself some grace because if they can't predict it, I'm not going to get it.

Jon Krohn: 00:53:04 Yeah, that's true. I do go around with kind of a lot of guilt of why can't I predict what's going to happen in AI better given this is what I'm doing all day. And then also another big guilt thing that I have, and this is actually going to lead into my next question, another big guilt thing that I have is that I always feel like I don't know enough. And so for those of our listeners who feel like they don't know enough about agents, maybe even particularly agent memory, do you have any particular resources that you recommend that they dig into to get started?

Richmond Alake: 00:53:37 Yeah. So one good spot, very easy, is the deep learning AI. Of course we did, memory aware agents, it covers everything within a good time, but also you can go to our Oracle developer center. The team is always putting out content. And then recently we put out a content called File System versus Database for Agent Memory. It was a good conversation a couple months ago and we put out some technical contents on there. So I mentioned free parts where you can get things, but I want to talk to feeling that it's very hard to keep up in this AI space. Everyone's feeling that. There is someone called Faith A Lee. She's known as the God mother of AI. And I was watching an interview with an interview at a conference and she said, she said herself, it's so hard for her to keep up with the space and she's the god mother of AI.

00:54:35 And I think things move so fast. And if anyone tells you they're an AI expert, they're probably trying to sell you something. But I would class myself as a lifelong learner in this space, right? That is what I signed up for and it



shows in how I'm teaching folks that I'm just learning and I'm teaching and I'm building and I'm just repeating that over and over again. This is our third episode, right? First one was over in computer vision and deep learning, and it's the same cycle. Just being in the arena, learning, building, educating. And that's what we're doing over in our course.

- Jon Krohn: 00:55:16 Yeah. You're always evolving and growing. That's for sure, Richmond. Do you think that that Architects of Intelligence book is your book recommendation for my listeners today? Oh,
- Richmond Alake: 00:55:25 We did book recommendations.
- Jon Krohn: 00:55:27 Yeah. We always have that. It's your third time here. You don't remember?
- Richmond Alake: 00:55:30 Yeah. Sure.
- Jon Krohn: 00:55:32 You already have that one, Architects of Intelligence. You can just lean
- Richmond Alake: 00:55:35 On that. Yeah, it's a good book. No, I want to lean on something better. There is a book called Thinking in Systems, and it's made by an awful called Donello. I do think, and it's not a technical book. I'll say it out loud right now, but I think it's important that us technical folks start reading non-technical books. And the books touches on how humans think in two systems, system one thinking, system two thinking, and one of them is the fast thinking and the other one is the slow. Yeah. System
- Jon Krohn: 00:56:03 One is the fast one.
- Richmond Alake: 00:56:04 System
- Jon Krohn: 00:56:05 Two is slow.



- Richmond Alake: 00:56:05 Exactly. And I think I was reading that book and I think it applies so well to what we're doing in terms of this agent, engineering and building memory systems. I do think, and I don't want to overuse the parallelism with human memory, but I do think that we can learn a lot from ourself because nature is the best architect of intelligence. So we can learn a lot from the way we process information. Another book that is on my ... So that's my book recommendation. Okay. It's Thinking and Systems, but another one that I think goes with it is-Bonus. I've not read it yet, but it's literally next on the list is The Organized Mind, right? Oh, yes. Yeah. Have you read that?
- Jon Krohn: 00:56:47 It's actually sitting ... So in my childhood bedroom, it's a house that my parents still have. Yeah. It has been sitting on the bedside table for years. And I'm always like, oh man, I can't wait to get to The Organized Mind.
- Richmond Alake: 00:57:04 So yeah, Thinking in Systems and the Organized Minds are two books. I don't know if you gave me the recommendation, but I was trying to come away from the non-technical side because I do that all day, every day, and just trying to go more in the abstract neuroscience, obviously, and just draw inspirations from there.
- Jon Krohn: 00:57:18 Well, yeah, I mean, it's interesting. I mean, I would say that both of those books are still technical. It's just that they're not software specific.
- Richmond Alake: 00:57:25 Yeah, they're not soft. Yeah.
- Jon Krohn: 00:57:26 Yeah. Yeah. It's not like you gave us a novel as a recommendation, which lots of our guests do.
- Richmond Alake: 00:57:32 Really? Oh, I'm boring.
- Jon Krohn: 00:57:34 Next time. Nice. All right. And then this has been an amazing episode, Richmond. As it always is with you, I enjoy our time together. I always learn so much. You've



already given us tons of resources for learning more about Agent and Agent Memory after this episode. What are also ways that people can follow you or Oracle on social media to get more information as it comes out after

- Richmond Alake: 00:57:59 This
- Jon Krohn: 00:57:59 Episode?
- Richmond Alake: 00:58:00 Yeah. So to follow me, just follow me on LinkedIn, Richmond Lake, and you'll find me also on Twitter or X, still experimenting with it, but LinkedIn is the best way to see what I'm talking about. And for Oracle, we have loads of ... Oracle is on all the social platforms. So Oracle developers on LinkedIn and also Oracle developer on X as well and on Instagram as well. So we are everywhere trying to educate, build, and yeah, we're in the arena.
- Jon Krohn: 00:58:27 Nice. Well, we'll be sure to have links to all of those in the show notes. And thank you so much for taking the time out of your time in London or out of your time in New York to join me here. Wow, that's a weird memory glitch or it's some kind of goddam. There's a memory conflict right there. Need a brain scan. Yeah. Fantastic. Thank you so much for taking all this time with us, Richmond, and educating us all on all these great things, agent memory related. Hopefully we'll be able to have you on
- Richmond Alake: 00:58:56 The show again. I'm looking forward to the next city. We did one online. Tokyo. We did one in London. We did one in New York. We're going to do one in Tokyo. Let's do that. Yeah. Let's
- Jon Krohn: 00:59:04 Do it.
- Richmond Alake: 00:59:05 Thanks for having me.
- Jon Krohn: 00:59:05 Perfect. Cheers. Always love having Richmond Alake on the show. This time around, he magnificently covered how agent memory is the encapsulation of systems,



embedding models, re-rankers, databases, and LLMs that allows AI agents to learn and adapt with new information over time rather than starting from scratch every session. How there are four types of agent memory drawn from human cognition, episodic, semantic, procedural, and working memory. He talked about how he advocates for a memory first agent harness and a key principle he emphasizes is reducing cognitive load for both LLMs and developers, for example, by consolidating into a single database that handles vectors, graphs, relational, and spatial data to avoid the anti-pattern of gluing together multiple databases. And he made a prediction that the flattening of AI engineering roles will happen in the near future where the future to developer will need end-to-end understanding of the full agent stack.

01:00:04 All right, as always, you can get all the show notes, including the transcript for this episode, the video recording, any materials mentioned on the show, the URLs for Richmond's social media profiles, as well as my own at superdatascience.com/985. Thanks to everyone on the SuperDataScience Podcast team, our podcast manager Sonja Brajovic, media editor, Mario Pombo, our partnerships team which is Nathan Daly and Natalie Ziajski, our researcher, Serg Masis, writer, Dr. Zara Karschay, and our founder, Kirill Eremenko. Thanks to all of them for producing another excellent episode for us today for enabling that super team to create this free podcast for you. We're deeply grateful to our sponsors. You can support the show if you're listening to this by checking out our sponsor's links, which are in the show notes. And if you yourself would ever like to sponsor an episode, you could find out how at johnkrohn.com/podcast. Otherwise, please help us out by sharing this episode with folks that would love to hear about agent memory, review the episode on your favorite podcasting app or on YouTube.



SuperDataScience

01:01:04 Subscribe if you're not already subscriber, but most importantly, just keep on tuning in. I'm so grateful to have you listening, and I hope I can continue to make episodes you'd love for years and years to come till 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.