**Transcript**

21 August 2025, 03:58pm

 **Michael "Newport Storm" Redlich** 1:00  
Hey, Marcus, how are you?

 **Markus Eisele** 1:02  
I cannot hear you, but I'm not sure it's me or it's you.

 **Michael "Newport Storm" Redlich** 1:05  
I just unmuted, so maybe you can hear me. Oh, yes, I can hear you perfectly.

 **Markus Eisele** 1:09  
Still can't hear you. Can you hear me?  
OK.

 **Michael "Newport Storm" Redlich** 1:16  
How's that? Any better?

 **Markus Eisele** 1:20  
Let me try this. Try again. Say something.

 **Michael "Newport Storm" Redlich** 1:23  
There's something OK.

 **Markus Eisele** 1:24  
Perfect. Now I can hear you. It was me after all. Oh man, getting things back up after two weeks of vacation is kind of a pain.

 **Michael "Newport Storm" Redlich** 1:32  
Yeah, yeah.  
Yeah, yeah, I hear you. Oh, yeah. My wife and I travel all that. We're we're we're both retired too. So. But yeah, so, so I do, you know, when we're away, we both do our thing in the morning and then we have the afternoon and evening, you know, to.  
Do what we need to do so.

 **Markus Eisele** 1:56  
Nice.

 **Michael "Newport Storm" Redlich** 1:57  
So good. Yeah, yeah. We're actually in Newport right now. So, Rhode Island.

 **Markus Eisele** 1:58  
But.  
Oh, OK, that makes sense.

 **Michael "Newport Storm" Redlich** 2:04  
Yeah. So, yeah, we just got here yesterday and this morning was leisurely. So, you know, I thought I joined the meeting and then we'll head out.  
Oh, here comes Neil.

 **Markus Eisele** 2:19  
OK, there's Neil.

 **Michael "Newport Storm" Redlich** 2:24  
There it is. There he is. What's What's going on?

 **Neil Patterson** 2:25  
How's it going?

 **Markus Eisele** 2:26  
Hanging in here.

 **Neil Patterson** 2:29  
Yeah.

 **Michael "Newport Storm" Redlich** 2:30  
Not, not, not to, not to brag, but, you know, here's our view. It's hard to see. Oh, there we go. That's the Narragansett Bay, you know. So, yeah, well, we like it up here.

 **Neil Patterson** 2:30  
Oh, just just had our.  
Oh, nice.

 **Markus Eisele** 2:43  
Nice.

 **Michael "Newport Storm" Redlich** 2:46  
No, uh, not.

 **Neil Patterson** 2:49  
Yeah. So you're taking a call from the the hotel room.

 **Michael "Newport Storm" Redlich** 2:56  
Yeah. So, right. You know, we got here yesterday. I just tell Marcus and it's been a leisurely morning, so I thought I'd join and then we'll head out, so.

 **Neil Patterson** 3:06  
Cool.

 **Markus Eisele** 3:07  
John, you made it.

 **Michael "Newport Storm" Redlich** 3:10  
Oh.

 **Markus Eisele** 3:12  
It's just curious to see what I'm what stupid stuff I'm actually going to talk about.

 **John Clingan** 3:16  
No, but I I do have a quick question. Are these meetings recorded? Oh, there it says recording. I I couldn't find prior meeting recordings.

 **Michael "Newport Storm" Redlich** 3:17  
Yeah.  
There it is.

 **Neil Patterson** 3:23  
Yes.  
Yeah, they'll they'll be on the GitHub in the GitHub repo as soon as I get a few spare moments to put them there.

 **John Clingan** 3:35  
OK, OK. Yeah, I cause I was checking the notes right in various places that I I didn't see or or the YouTube channel, I I didn't see any any recordings. But oh, it's awesome that these are being recorded.

 **Michael "Newport Storm" Redlich** 3:35  
Hang on.  
M.  
How you doing, John? Long time no see.

 **John Clingan** 3:51  
It's been a while, man. You've you've you've been negligent in your micro profile meetings there, Mr. Redlich.

 **Michael "Newport Storm" Redlich** 3:52  
Yeah.  
Guilt. Guilty as charged. Yeah, yeah. But that's. Thank you. I know Tuesday. I totally forgot that we're back to the weekly meetings, but we were only home for two days.

 **Neil Patterson** 4:01  
Mhm.

 **John Clingan** 4:02  
Well, you're retired, so you're forgiven.

 **Michael "Newport Storm" Redlich** 4:15  
days and we had a new fence put in uh so that kind of kept me busy um and it's kind of funny the well not funny really uh we we marked off the property line and put a stake in the ground totally forgetting that our alarm service comes in and they  
hit the stake and it went into our brand new sighting. When flying. So, you know, the zero turn. Oh oh my god. So yeah, anyway, but these things happen.

 **John Clingan** 4:41  
M.

 **Neil Patterson** 4:46  
Yeah.  
So I I'm not sure how many we're going to get today because it's peak holiday season for many. So you know we it is recorded. We'll we'll post a recording and send it out. So Markus, why don't you take it away. You know hopefully we'll get a few more joining as we go through.

 **Markus Eisele** 5:07  
Yeah, give me a second. Let me just try to close some screens here and see how all of this works.  
I'm just not back into my most favorite working mode here, but we'll get there. We'll get there. So I think it was you, Neil, that invited me to talk and lead this discussion today. Let me see if I can get this full screen so you guys see this little.

 **Neil Patterson** 5:21  
Ha ha.

 **Markus Eisele** 5:35  
A little bit better and I was, I was actually thinking because the whole topic of AI obviously moved me since a while and the main reason is we kind of committed to a second book after the first enterprise job.  
Java book, we really wanted to see how we can actually put a little bit more up-to-date technology out in the field and we've been looking at a ton of engagements that we had individually as a team Speaking of developer.  
At IBM and Red Hat, obviously, and also us running all our own experiments because we're just curious and crazy and we wanted to figure out stuff, right? So at the end of the day, this will be the result that is coming out in November. You get a very.

 **Neil Patterson** 6:22  
M.

 **Markus Eisele** 6:34  
Very raw early draft from the usual places, and I'm not going to advertise any further, but we'll end up having 400 pages. Roughly, I'd say 1/4 of it is just terminology and putting things into context.  
Including architecture elements, platform elements, what's necessary. But we're also digging into all kinds of frameworks, looking at what Java developers can actually do with them. And this is not even exclusive to Quarkus. Quarkus is definitely one of the examples.  
We're showcasing, but yeah, that's kind of the why me, right? And the the next piece was I thought about why now?

 **Neil Patterson** 7:14  
Mm.

 **Markus Eisele** 7:20  
Why a I? I mean, obviously Java and the complete Java platform is really mature technology. It is something that I'd say 99% of the really large enterprises base their day-to-day business on.  
It's also something that we've seen being kicked off outside of our realm, like with all the GPT inventions and the Java ecosystem kind of managed to align forces we've seen.  
Lang Chain 4J coming to life. We've seen Spring AI, we've obviously seen the Lang Chain 4J integration into Quarkus and many other examples. I have a have a list, at least an incomplete but longer list of this. And we're also seeing a ton of business pressure. Customers are actively asking.  
About these kind of technologies, what are you guys going to going to offer me? Like how can I do X&Y and Z and how can I do this with Java? Because I really don't want to put prototypes and pilots and Python notebooks into production as.

 **Neil Patterson** 8:24  
Yeah.

 **Markus Eisele** 8:29  
Awesome as the 1st result is like at some point this stuff needs to scale. So the natural result is Java is kind of the the enterprise backbone for AI adoption because.  
If a I isn't getting closer to Java or vice versa, there won't be large adoption. It will be pockets, it will be interesting pieces, but we really need to make sure that all the enterprise requirements.  
Are able to be fulfilled with these kind of a I approaches and I think this is the exact reason why now. So Java already saved a couple of things in the past and this is kind of a provocative statement, but.  
Looking at the specifications, the the really provocative question that comes up is, is the platform and are the specifications too heavyweight for this kind of fast moving field. We've seen similar occasions and I'm just briefly mentioning microservices here.  
Where we've struggled to get a very long lasting and established corpus of specifications to move into a more lightweight area. Is that another kind of thing that we need to look after?  
So where's the action happening today? Obviously pretty much everywhere and this is all Java. So we have something that does LLM orchestration, so large language models, so text based stuff. We do have some initiatives around runtime interference.  
And actual GPU usage. We have already a ton going on within Jakarta EE slash Microprofile and obviously there's also this really long list and I'm only mentioning 3 here of established Java based machine learning.  
Libraries like suites almost. If you're looking at SMILE, what you can do with SMILE is is impossible, unthinkable if you start unraveling all these layers of modern artificial intelligence overview architecture pictures.  
This is the basis where a ton of this is basically built up on on top, right? So and leading into the uncomfortable pieces, it's ultimately a question of how can this influence standards and specifications?  
How would that look like? Do we do we actually need less explicit specifications? If we have something that is writing all the boilerplate, do we rush an API?  
Or wait or use something that is already established out there? Do we need new infrastructure components? And Vector DB is just one fancy example. How can we make sure that these new developer faced toolings that are emerging?  
Cloud desktop, whatever coding assistance you call them, actually do understand the specifications and the examples and how to implement them. So is that a new kind of tooling, new category that we need to define inside?  
The platform. Very uncomfortable questions to ask. Um.  
And there's also features that we could actually bake into the specification or the runtimes, I should say the implementations. Actually, we could do something like an A I Butler and I did a really teensy and lightweight proof of concept here. I'm not sure you guys have seen.

 **Neil Patterson** 12:17  
Mm.

 **Markus Eisele** 12:31  
We've seen the Quackos initiative, what they've been baking into the developer console that is much more advanced. So this is really just a dumb, stupid example. But even that ultimately already gives developers a hint.  
And if it would be using something that is a little bit more advanced than just a pre-trained and ready to pull down model like I did in this example, it could actually help way more than just give a a suggested fix rate.

 **Neil Patterson** 13:02  
Mhm.

 **Markus Eisele** 13:06  
OK, what's coming? I'm assuming we'll see a ton more AI influence on the platform language, the language platform. Just think about the Vector API Babylon. Potentially we'll see a ton of movement in Graham V M.  
I should mention Tornado a little bit, but yeah, let's just keep it there. I heard that the first specifications maybe around micro profile might be touching a I very soonish and maybe that is something like telemetry. Emily will definitely know.  
No more. I don't. There's also some talks around standardizing tools and obviously if we're looking a little bit farther out, the market assumes that half the code bases out there that we're seeing and dealing with will be generated.  
And that ultimately has a big influence on what specifications actually do and how they help safeguarding these generated code bases. So let's discuss what do we need?  
What don't we need? Where are we? What are you guys seeing?

 **Neil Patterson** 14:24  
So one one of the questions I think that comes to my mind immediately is moving like this fast with so many different areas. It sounds to me we're going to be building up a lot of technical debt.  
And how do we, how do we move that? How do we avoid that if we can? But how do we make sure that customers that adopt early with some of these things aren't burned by having to rewrite?  
Um.

 **Markus Eisele** 14:59  
Good question in the area of move fast and don't break things. This is definitely an interesting thought because as I mentioned, it's not that we don't have any users out there, it's it's a ton of enterprises who rely on these specifications.

 **Neil Patterson** 15:14  
Mhm.

 **Markus Eisele** 15:19  
For various reasons. The reasons might have shifted in the last 15 years, but it is still something that where knowledge is comparably easy to source, which is taught in universities, has a lot of knowledge out there in the field, is really good fit for business requirements, so.  
Can we break this? Can we make this fit in a way that it still stays stable? Good questions, absolutely.

 **Neil Patterson** 15:45  
Right.

 **John Clingan** 15:47  
Yeah.

 **Michael "Newport Storm" Redlich** 15:47  
Yeah, what what are the interest?  
Go ahead, John. Yeah.

 **John Clingan** 15:53  
Sorry, right. Well, I was just gonna kind of follow along, you know, with this balance of moving fast and not breaking things from a micro profile. Well, actually, you know, just from a a specification perspective, right? I I think the whole value of specification is that you don't break things.

 **Michael "Newport Storm" Redlich** 16:01  
Mhm.

 **John Clingan** 16:12  
Right. Which is why when we first started with A I in in in the micro profile community, you know there was some discussion of creating specifications and I was kind of against that because it moves so fast and it obviously will break things just because it's moving fast.

 **Michael "Newport Storm" Redlich** 16:14  
Yeah.

 **Neil Patterson** 16:15  
Mhm.

 **John Clingan** 16:33  
And even within micro profile, there isn't consensus on the pace of change, right? So some folks think, well, the point of micro profile is you create a right backwards incompatibility.  
Isn't as strong a notion in micro profile as it is in Jakarta, right? In micro profile, we are willing to break things. Now things have changed with Jakarta right as they've been deprecating more right to get rid of some of the older, less relevant.

 **Michael "Newport Storm" Redlich** 16:58  
Mm.  
Mhm.

 **John Clingan** 17:07  
Specs or or features, but I still think that users do expect some sense of stability within specifications and.

 **Michael "Newport Storm" Redlich** 17:22  
8.

 **John Clingan** 17:23  
Sorry, I'm just kind of dumping thoughts. The the one area where I I could see or the one concept I could see where breaking changes are, I don't know, acceptable or not is specifications.  
These days, in my opinion, are less about portability and more about ecosystem. The strength of an ecosystem right where Jaxx Rs or you know, JPA or whatever, right? Once I know them then.

 **Michael "Newport Storm" Redlich** 17:54  
OK.

 **John Clingan** 17:57  
I can write them in pretty much any context, whether it's it is Quarkus or Liberty or Helidon or whatever, right? Whatever the runtime is, it's not really about being portable, it's just about having a lot of investment in a single area where I have a set of skills that I can transition.  
From one job to another, one product to another if I need to, but it's less about application portability. So being able if you break changes once, you break them everywhere. Sorry, if you break things once, you break them everywhere I guess. But the concept of a consistent skill set still remains. I guess I'll.

 **Michael "Newport Storm" Redlich** 18:30  
Yeah.

 **John Clingan** 18:36  
I'll pause there and and see what other folks thoughts are.

 **Michael "Newport Storm" Redlich** 18:39  
Yeah, there's a lot going on here, of course. And and I know it was Ivar who said that the hype of a I was starting to come down a little bit. This was at last year's trends report that.

 **Markus Eisele** 18:40  
I I couldn't agree more.

 **Michael "Newport Storm" Redlich** 18:58  
That info cube puts out. But the other thing, it was kind of interesting. There was a debate back at Dev Nexus in March. I don't know if any of you guys were there for that, but this was at the summits, either the JUG summit or the.  
The Java Champion Summits about.  
About Java in general, you know not not getting involved in creating LLX, let let the Pythons and all the other. I thought it was interesting seeing Cobol there, but you know, but then you know since Java and Jakarta are in enterprise.  
And good at enterprise, yeah, we could connect to those technologies and.  
You know, and and use a I for, you know, for what people want to do. So I don't know where where all that is. And I guess Marcus, I think I heard you say something about people are asking.

 **Markus Eisele** 20:00  
Yeah, I think.

 **Michael "Newport Storm" Redlich** 20:00  
for us in the Java community to do these things, so yeah. LLM, yeah.

 **Markus Eisele** 20:04  
What what you folks also what what you should at least keep on the radar is that I I mentioned Java running a very large amount of enterprise workloads that implicitly also means.  
That the domain logic and the data is sitting in those systems. And guess what you'll need to make LLMS useful in a specific context is exactly that. So if we're saying we're not getting involved with training, whatever.

 **Michael "Newport Storm" Redlich** 20:27  
Mm mm.  
Yeah.

 **Markus Eisele** 20:40  
Whatever that has a direct implication in the means of we're cutting ourselves out of the data gold for for this kind of scenario, yes, you can.

 **Michael "Newport Storm" Redlich** 20:46  
You.  
Yeah.

 **Markus Eisele** 20:55  
Absolutely argue about putting all this not in from the back on the training side, but also from the front end on the like argumented piece.

 **Michael "Newport Storm" Redlich** 21:03  
Mm.  
Mm.

 **Markus Eisele** 21:09  
Question is, how long will that hold? It takes a lot more to be useful.

 **Michael "Newport Storm" Redlich** 21:10  
OK.  
Right, yeah.  
Yeah, and and you know, I I'm not sure what side of the fence I really agree with, you know, with that debate to be honest. And and perhaps we should be involved in in creating those things, you know, so oh, Majid asked about Gentec, so.  
In the in the in the comments, but that's that's a good one too. I know IBMS involved in that. But anyway, yeah, yeah, there's there's there's a lot going on here and I I suppose ultimately there should be some kind of standardization.  
Like with everything else.

 **Markus Eisele** 21:52  
I I think we we need to have an opinion how we want to deal with this and this is true for both Jakarta and and micro profile and I'm personally for my own projects leaning towards.

 **Neil Patterson** 21:52  
Yeah.

 **Michael "Newport Storm" Redlich** 21:54  
Yeah, yeah.  
But.

 **Markus Eisele** 22:09  
Making sure I have the viable use cases that basically influence the level of AI I'm using for certain scenarios. So I'm not bluntly trying to put it out everywhere, I'm just trying to be.

 **Michael "Newport Storm" Redlich** 22:21  
Mm.

 **Markus Eisele** 22:28  
Very prescriptive where it actually makes sense. And another piece is also super interesting. Frank Greco keeps reminding me about this and I'm very thankful because a I is not just large language models, there is a ton more that is.

 **Michael "Newport Storm" Redlich** 22:41  
OK.  
Yeah.

 **Neil Patterson** 22:45  
Yes.

 **Markus Eisele** 22:47  
Traditional machine learning, which is something that also still has very high value to a lot of customers. So it's not the the regular customer that is looking to implement a chat bot or any kind of content generation strategy.

 **Michael "Newport Storm" Redlich** 22:47  
Yeah.  
It.  
Mhm.

 **Markus Eisele** 23:06  
They're more interested in finding new ways to interpret the data they are having with modern slash established technologies. So I think what?  
What would be a first good step is to decide about tooling support for the end users of specifications, which I consider developers to be. So what can we actually deliver? James Ward wrote a teensy little MCP server on top of his Java doc web.

 **Michael "Newport Storm" Redlich** 23:31  
Mm.

 **Markus Eisele** 23:40  
Side. So that that is a very simple example because all the stuff that's in Maven Central already can be exposed via that, but there's there's more potential in teaching coding assistants to.

 **Michael "Newport Storm" Redlich** 23:41  
Mm.  
Hmm.

 **Markus Eisele** 23:56  
Be aware of specifications and use them wisely. That is 1 bucket. The other bucket is on what to rely to abstract what kind of A I technology. That is another piece and that could be very strongly guarded by use cases.

 **Neil Patterson** 24:00  
Mm.

 **Markus Eisele** 24:15  
In my opinion at least. So we need to figure out where the the real desire is. Are we talking about pulling in more?  
Low level platform concepts, vector APIs, math libraries. Do we need a stronger foundation that allows us to do stuff? Do we need abstractions?  
And one of my favorite examples is the HTTP session. We've done this and it was, I'd say, one of the success stories of enterprise Java application service. You were able to literally store something and accesses on all.

 **Michael "Newport Storm" Redlich** 24:43  
Mm.  
Mm.

 **Markus Eisele** 25:01  
Application server instances across the cluster. So guess what? Agents and LLMS and stateless interactions with restful endpoints. There is no such thing as a session, a user context. There is no such thing as memory.  
So do we need higher level abstractions on top? Langchain 4 J has an approach. You could think about other examples like caching. I've also heard people talking about JPA for LLMS, which I'm not a big fan of, but anyway, so yes, the next bucket are.

 **Michael "Newport Storm" Redlich** 25:37  
Thank you.

 **Markus Eisele** 25:40  
And after all of that, we might actually be able to move into something that ties this down in terms of long term supportable specifications, right?

 **Neil Patterson** 25:53  
So do do you see the potential for there being something like a an A I profile that would contain a set of specifications that were tailored or?  
Ideal for various AI use cases.  
Would that be something we could think would happen in the future or?

 **Markus Eisele** 26:20  
So to two pieces, I am personally a big fan of profiles. I always thought this is a very strong concept to bring down the total size of specifications and platforms.

 **Neil Patterson** 26:31  
Mhm.

 **Markus Eisele** 26:36  
So yes, on that side, is it time for an AI profile? I don't know. Is it time for a way to define an emerging space as a profile with different?

 **Michael "Newport Storm" Redlich** 26:43  
Mm.

 **Markus Eisele** 26:52  
Release cadences with different specification requirements and boundaries maybe. So I think we cannot drop the ball, but we also need to be sure to not artificially invent something that is just not being seen as a standard.

 **Michael "Newport Storm" Redlich** 26:54  
Mm.  
Mm.

 **Markus Eisele** 27:11  
Out there right now. So giving us enough levay, experimentation, room to breathe by still making sure we are on this topic, on top of the topic, and we're communicating this correctly outside like it is of interest. It is something that we deeply.

 **Neil Patterson** 27:12  
Right.

 **Michael "Newport Storm" Redlich** 27:28  
Yeah.

 **Markus Eisele** 27:31  
Deeply care about. Cannot talk in the we because I'm a person. I'm Marcosier in no official capacity. Just want to have this on record.

 **Neil Patterson** 27:32  
Right.

 **Michael "Newport Storm" Redlich** 27:39  
So, so Neil, I assume you meant an AI profile to complement the core and web profiles in Jakarta EE. Yeah, OK, yeah, just making sure.

 **Neil Patterson** 27:40  
Right.  
I I.  
Yeah, I I mean if if I if I think longer term, yeah. If I think longer term, you know we're having this debate as well about you know what does the there's core profile which kind of well aside from micro profile, core profile kind of broke the.

 **Michael "Newport Storm" Redlich** 28:03  
Yeah.

 **Neil Patterson** 28:06  
The mold, so to speak, of always needing to have like platform, right? And so could we, I'm I'm thinking to be more agile and you know, even to the point of releasing specifications out of a time given release cycle.  
Having profiles that would be associated with technologies or with industries, that kind of makes sense to me because that would enable us to be more agile, more responsive.

 **Michael "Newport Storm" Redlich** 28:37  
Mhm.

 **Neil Patterson** 28:42  
To emerging things like AI.

 **Michael "Newport Storm" Redlich** 28:45  
Mm.

 **Markus Eisele** 28:46  
Again, I personally am a big fan. There's a couple of historic things and decisions that I do not understand, but I don't have to. So again, I I like the thought of having the ability to offer something that is not just technically.

 **Michael "Newport Storm" Redlich** 28:49  
Mhm.

 **Markus Eisele** 29:03  
That restricts in many places also having the opportunity to just offer and I like the the vertical idea, the the domain specific profile. Even if I think about the past where I had to deal with a ton of like finance, math and all the long-running.

 **Neil Patterson** 29:15  
Mhm.

 **Michael "Newport Storm" Redlich** 29:16  
M.  
OK.

 **Markus Eisele** 29:23  
Transactions and insurances. That is all very specific. You can build this. There's a ton of abstractions that are potentially reused and rebuilt in many, many companies. So yeah, I'll shut up, John.

 **Neil Patterson** 29:25  
Mm.

 **John Clingan** 29:39  
Yeah. So, so interestingly, the idea of an A I profile, which I haven't thought about, this is interesting, right, that there was a, you know, that Michael brought up. So one of my initial thoughts behind the core profile, you know, when we were.  
Starting to talk about it was that it could move at a different pace because micro profile moved at a different pace than Jakarta, right? So the interesting thing about profiles is that is that it's not just a domain kind of separation potentially, but it's also a release cadence difference. So AI is moving.

 **Michael "Newport Storm" Redlich** 30:03  
Mm.  
Yes.  
OK.

 **Neil Patterson** 30:16  
Yeah.

 **Michael "Newport Storm" Redlich** 30:18  
Mm.  
Mhm.

 **Neil Patterson** 30:34  
Yep.

 **John Clingan** 30:34  
File than the full platform, but it doesn't, you know, but they still tend to be thought of monolithically, right? Where one is released in cadence just so people can show progress and stuff. But AI could be one where it's actually more detached from the other.  
Profiles, so it could. It could literally move at a faster pace than the other ones. OK, I'm done.

 **Neil Patterson** 30:58  
Right.

 **Michael "Newport Storm" Redlich** 30:59  
Yeah, like I I I actually just just as a an aside, I'd like how micro profile started out with the three original JSR specs and you know CDI and I think Jax RS and and Jason P and then.

 **Neil Patterson** 31:00  
I like that, yeah.

 **Michael "Newport Storm" Redlich** 31:17  
You know as the core profile emerged now you know those 3 specs which were the original three were in the core profile and then now micro profile now used core profile as you know as that so I but.  
I don't know, I'm just weird that way sometimes. But again, you know this this ties into a having an AI profile that at some point but.

 **Neil Patterson** 31:45  
Yeah. Now Majid did ask that I I think AI can kill the UI because I think you were talking earlier about different ways of using AI. What's your thought on that, Marcus?  
Does AI take over and get rid of the need for a UI?

 **Markus Eisele** 32:07  
You all have played with it. You all have seen there are better results and worse results. I am not sure personally as it stands today.  
I'm also not so sure about future capabilities. There is a ton that is possible if you would duct tape stuff together underneath and and pull it into a product. And I'm randomly mentioning cause I hear I could be mentioning every.  
Other IDE they all kind of follow the same and if you even want to look at source code you can. The Microsoft copilot is is open source in the meantime, so this is all just prompts plus magic I.

 **Michael "Newport Storm" Redlich** 32:53  
M.

 **Markus Eisele** 32:58  
I I don't know, man. I rubbed my belly. It didn't tell me.

 **Neil Patterson** 32:59  
Yeah.

 **Michael "Newport Storm" Redlich** 33:04  
Yeah.

 **Neil Patterson** 33:04  
Right, right.  
Yeah, I I I'm interested in knowing your experience with Langchain4j, because that's outside of our kind of purfew around Jaccardi and Microprofile it. What is, what do you see the long-term relationship there being?

 **Michael "Newport Storm" Redlich** 33:13  
Yeah.

 **Neil Patterson** 33:24  
Are being for that.

 **Markus Eisele** 33:26  
I don't know. Is there a similar reference case that we could mention? I I don't know. No idea. Totally, totally blunt, I mean.

 **Neil Patterson** 33:35  
Yeah.

 **Markus Eisele** 33:38  
It is kind of the glue, so a very interesting glue with all the relevant pieces. I seriously no opinion right now.

 **Neil Patterson** 33:50  
OK.  
This is the first time I've kind of seen that as being, you know, reliant on some other area, some other set of specifications, if you will, that we haven't.  
really been involved with before to any

 **Markus Eisele** 34:11  
Hm.  
I mean that could also be a proposal going forward, right? Redirect some of the energy and initiatives and develop in the open with the idea of pulling out a specification that is basically.

 **Neil Patterson** 34:30  
Mhm.

 **Markus Eisele** 34:32  
Openly I and I'm again, no official capacity, not Red Hat, not IBM tonight, but just my thoughts. I'm pretty sure somebody will hate me for this. But anyway, I I love the idea of proving value before standardisation. Standardisation always has that feel and smell to it that it stalls.

 **Neil Patterson** 34:38  
OK.

 **Michael "Newport Storm" Redlich** 34:42  
Yeah.

 **Markus Eisele** 34:52  
Innovation. We've all had that discussion in various capacities. Yeah, definitely. I mean, you get more people relying on this, so it's a really good idea to prove something's value before you actually tie it down.

 **Neil Patterson** 34:54  
Or restrict.

 **Markus Eisele** 35:09  
So that also would give you a good opportunity to basically have a reference implementation ready. Um.

 **Neil Patterson** 35:15  
Yeah.  
Very, very true. OK. Well, I appreciate that we're at the bottom of the hour and I really appreciate you coming on today and sharing with us. I.  
We'll share the the recording so we get more eyes on it and hopefully that'll generate some discussion going forward as well. But you're always welcome back, Marcus, John, don't be a stranger and.  
Majid, good to see you again. And Mike, of course, partners in crime in terms of agendas and stuff, so.

 **Michael "Newport Storm" Redlich** 35:57  
It's.

 **Markus Eisele** 35:59  
Mike, enjoy your time over and good good luck with whatever you guys are up to.

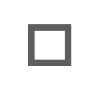
 **Michael "Newport Storm" Redlich** 36:04  
But.

 **Neil Patterson** 36:05  
All right, great. Thanks all.

 **John Clingan** 36:06  
Yep, good. Yep. Bye.

 **Michael "Newport Storm" Redlich** 36:07  
Yeah. Thank you. All right. Thanks, Marcus. Take care, guys. Good to see you all. All right. Bye.

 **Markus Eisele** 36:07  
Talk soon. Bye.

 **Neil Patterson** stopped transcription