

The Open Source Way

Episode 18: Eclipse Dirigible – An Open-Source Platform for End-To-End Rapid Development of Business Applications



Transcript

Karsten: Welcome to the Open Source Way. This is our podcast, SAP's podcast series about the difference that Open source can be. In each episode, we will talk to experts about Open source and why they do it the open source way. I'm your host Karsten Hohage, and today I'm going to talk to Yordan Pavlov and Dragomir Anachkov about the project Eclipse Dirigible. Hi Dragomir, and hi Yordan.

Dragomir: Hi Karsten.

Yordan: Hi, happy to be here.

Karsten: Great to have you here. Now, Yordan is a project lead of Eclipse Dirigible and a solution architect for the project XSK. Both are open source projects. He's also one of the major contributors on the two GitHub repositories. Yordan joined SAP in 2013 and was part of the core teams working on what is now known as SAP Business Technology Platform. Recently, he switched his focus, mostly toward helping SAP customers and partners to migrate and develop their solutions to SAP BTP. Dragomir has been a user assistance developer at SAP since 2016, I think. Apart from writing documentation, he's also an avid video creator. He has been working on the Eclipse Dirigible project for the past four years by contributing to its documentation and the YouTube channel. So where are you located right now? Dragomir, are you at home?

Dragomir: Yeah, exactly. In Sofia, Bulgaria. And I think Yordan is there as well. I mean, here.

Karsten: You're both not in the office and are working from home, as most of us do these days.

Yordan: No, no, since the start of the pandemic, we've been in home office mode, so it's been almost two years.

Karsten: Wow, you've been constantly only working from home all this time?

Yordan: Yes, since March 2020 I've been working from home.

Karsten: Wow, okay.

Yordan: It's kind of nice, you know, you get used to it.

Karsten: I never really got used to it. That's why I have to confess I'm in the office today. I do this like once or twice a week, but I don't know. Here in Walldorf, in the Walldorf buildings, at least, I'm not aware of anything happening. But we don't want to be talking about the pandemic for too long, I guess. Let's get to what we're here for: Eclipse Dirigible. Can either of you tell everyone, in short, what that is?

Dragomir: Yeah, I can briefly talk a bit about the project and what it represents. Eclipse Dirigible is a platform that provides development tools and the runtime environment to developers. It has its own Web IDE, and it also provides an in-system development experience based on JavaScript, where we provide our own set of JavaScript server-side APIs. It's quite different from what Node.js has to offer, and the same system development experience is similar to what we've had at SAP for many years. And it's also known as the ABAP programming portal. At Eclipse Dirigible, we support both the standard programming model, which basically means that developers code their apps manually from start to finish, and we also support the low-code/no-code programming model, which means that we provide developers with the appropriate tools to help them code their applications faster and more easily.

Karsten: Okay, so most of all, it's a development environment, right? And it has this special relationship with what we also mentioned at the beginning: XSK. What is XSK, and how does it relate to Dirigible? Maybe Yordan at this point.

Yordan: So, XSK; first, let's start with what XSK proposes, so it's a kind of compatible runtime environment and also development environment for XS classic applications. So, these kinds of applications were on the SAP... – now known as SAP's Business Technology Platform. But back then it was "SAP Cloud Platform", and it was some kind of development model that was done in the HANA system itself. So, it has the same system development capabilities to offer. And what we're currently offering with the XSK

open-source project is a way for our partners and customers to continue using their applications, but to transfer them into the new cloud environment that SAP has to offer, which is based on Kyma and Kubernetes. So, basically, this is what makes this case and the relationship between Dirigible and XSK is very special because XSK actually uses the Dirigible project as a foundational layer, so, we can assume that 90 percent of what XSK has to offer actually comes from Dirigible.

Karsten: I don't know if I understood that correctly. XSK is basically an environment where you can pretend that the HANA XS server is still being supported, which has actually been deprecated a couple of years ago, right?

Yordan: Exactly, exactly. So, it's kind of emulates the application, and you don't have to rewrite them from scratch. Just basically keeps the investment of partners and customers.

Karsten: Okay. Keep the investment, that's a good keyword, I guess. Now both these are separate projects, right? There is the Dirigible open-source project and the XSK project which are separate, right? But XSK basically runs inside Dirigible. Could we say it like that? Okay. And they're both fully and truly open source. What do we need to note about them as far as that's concerned?

Dragomir: Yeah, they are both truly open source. Eclipse Dirigible is an open-source project, part of the Eclipse Foundation since 2015. It is a founding member of the Eclipse Cloud Development Working Group, along with Eclipse Che and Eclipse Theia, among others. Eclipse Dirigible started internally at SAP more than ten years ago, and during this period of time, it went through several major updates, and its latest version of 6.0 includes own system development and local features that the developer might need. And you can try out Eclipse Dirigible by going to our official website dirigible.io. On the homepage, you can click the "Try it out!" button and you can immediately start a trial of Eclipse Dirigible, the Web IDE...

Karsten: And is it also open for anyone to contribute, or is it a largely SAP driven project?

Dragomir: Everyone's free, you're all free to contribute. You can just go to our GitHub repository. There is a link on the homepage of our website that leads to our repository, and there you can post issues, ask questions, and there is also a discussion step there. So, yeah, feel free to join in and ask questions and contribute. We're happy to have you.

Karsten: Okay, great. And then, now that we know that it's open for everyone in a little more detail: How exactly is it used? Is it a totally independent runtime environment, or do you plug it into something else? What are the use cases? Just a little more detail.

Yordan: You know, I could spend like an hour talking about the use cases and the things that it can be used for. But briefly, what I can say is that it has a runtime environment, so something like applications that are like Tomcat, and it has a design environment, so it has basically a Web IDE. Of course, when you package your production application, then you can just eliminate the Web IDE and package the application along with the runtime environment. So, there are two flavors of application development. So, first of all, Eclipse Dirigible is for application development, mostly for business application development. Yes, it can be integrated with the services that are coming from the cloud platform, but also, at the same time, Dirigible can run locally, so, you can think of something that can even be run on premise instances. So, this is important for SAP customers, as well. Or, on the other hand, it can just use some of the services that are coming from the cloud platform like SAP HANA Cloud instances or, for example, Postgres or even some messaging and whatsoever. So, this is how you actually install it, deploy it. But from the application point of view, you have just two flavors of application development: You have the pro-code application development guy and the low-code application development guy, and the first guy can basically write manually whatever he wants, in the way he wants, and we give him the tools that he needs. So, you have an editor tool that you can use to edit your code. Basically, we support server-side JavaScript code to be written. It is actually important to say that it is not Node.js, and we're using both open-source projects under the hood grout-js and grout-vm that are coming from Oracle to execute server side JavaScript.

Yordan: And what is also important to mention here is that it is synchronous JavaScript. You don't have just callbacks in async awaiting that you have in Node.js, and it is smooth threaded. So, we basically leverage what Tomcat provides us, so, it can scale many, many threads at the same time, and you can just write the code synchronously.

So, what there is to mention is that it's actually based on Java, and it requires Java to run. And maybe a smaller portion of the use cases is to use only parts of Dirigible or other frameworks and to embed that part in your application. So, for example, we have one module that can be embedded and could actually replace JPA. So, it's kind of lightweight persistence major. It's for Java applications to be embedded. We have a similar framework, which actually provides all data capabilities again for Java, and it can be integrated with the codebase for change management. So, all that can be used separately from Dirigible and for the local application development. Maybe what I have mentioned is that you have these drag and drop capabilities to model your application. So, basically to model your domain model to have a relationship between entities, then to, for example, select one of the entities, provide some settings, like, what should be the user interface, what should be the widget there, whether it's textbook or a dropdown or a number or whatsoever. And then with a few clicks and with a few methods that you need to import, the complete application is basically generated. By the way, you can provide your own set of templates, so it's pretty extensible. So, that's it in short.

Karsten: Okay. That was the medium long version after the short one in the beginning, I won't even pretend that I understood everything you said. I understood some parts at least, but now a little bit of a different perspective on that: Why would I use Dirigible? I mean, apart from it supporting the programming model, why would I use Dirigible instead of another Eclipse-based application development environment?

Yordan: Yeah. So maybe what I didn't mention in the beginning is that Dirigible is not a general purpose Web IDE. So, for example, you have the Eclipse IDE, the most popular one, the Eclipse Theia, the Eclipse Che Web IDEs. So, on the SAP side we also have this business application studio. But most of them, they actually provide a generic way of how to develop applications, so, you can write in Java and JavaScript, in Python or whatsoever. And here in Eclipse Dirigible, what we strive to achieve is actually to provide a tailored environment for application development. We promote only this JavaScript server-side language that we have as a target for the application development. We have the tooling that you need to actually develop applications, or we have a set of prepared perspectives that you will be using during the application development. So, it's a tailored environment for exactly a few kinds of applications that need to be developed, and most of them are really tied to the needs that SAP customers and partners have so far.

Karsten: I seem to remember when we talked before this recording here that you said something about Eclipse Dirigible being the only programming environment supporting especially the... How did you call it?

Yordan: Low-code development...

Karsten: The low-code...

Yordan: So, basically, Dirigible is one of a kind. So, it's open source, it provides low-code capabilities. So, you have many tools nowadays that provide these kinds of capabilities, but I would say all of them are actually paid. You have to have some contract to use them and to pay for them on a regular basis, and Dirigible is actually open source. You can just take it and use it, and if you want, you can extend it like we did with the XSK project itself. Of course, there is other projects that provide local capabilities like what we have in the Eclipse Foundation, but they're mostly towards the desktop applications and they're actually not IDEs, but some kind of frameworks, and they don't run in the browser. So that's what distinguishes us from the competition. So, we run in the browser, we have Web IDEs, we have piloting, we provide local features, we're open source.

Karsten: But that also means that it's not limited to just servicing the XS model.

Yordan: Yes, of course.

Karsten: Okay, so what other models or programming styles does it support specifically?

Yordan: So, for example, what I mentioned for these drag and drop capabilities is actually known as model-driven architecture. So, you have the domain model for applications, you can model the entity, so, you can just drag and drop entity and add fields for it, like first name, last name, name, whatsoever, then add drag and drop, another entity, then link them together. All of this happens in our visual editor, of course. So it's really nice to have the entities there, so you can actually visualize it. So, for example, in the Cloud Application Programming model you have text definitions only. Of

course, they could be very complex. But what we are missing there right now is the ability to see the domain model. So, just glimpse on it and say, okay, I have the customer, the partner, the sales and whatsoever entities there, and I want to extend that one, or I need to change that one, you know, so, you really have to write the text. So, it's kind of reading an article versus watching a video on YouTube. So, it's that kind of difference.

Karsten: Okay.

Yordan: So, this is for the model-driven architecture, of course, another style of the local, because in the local application development we have many styles, and with some colleagues, we determined that there are up to 44. By now, there are probably more than 60. So, you have very different flavors like, say, NDA, which is model driven architecture, which is business process modeling. You even have AI-assisted development. You have this RPA, which is Robotic Process Automation, and many, many other flavors. So, I don't want to list them all. And the other major flavor that we support is the business process modeling, so you can model the steps for one business process. So, you can imagine, for example, a checkout process; so, you can model all the steps of the checkout, for example, the validation of the card, then the transaction itself, then sending a notification to the shipment department to place the shipment. And all that could be modelled as a process because it's a complex thing and the processes are actually one of the things that SAP is famous for.

Karsten: sounds interesting. Let's maybe get back to the to the XS part of it, because that, of course, is probably a very interesting thing for longtime SAP customers who have been doing things on XS in years before. We already said it basically is able to emulate XS still being around, right? Can you just tell us the story, maybe? What exactly was XS and how does it come to the necessity to emulate it these days?

Yordan: Okay. So, to tell the story of XS I have to go back in time, like 10 years. So, it started even before Dirigible here started. So, it is a proprietary development model that SAP has to offer. It was in the beginning of 2010, 2009, something like that. So, back then it actually started. It has the goal to be able to place your application code closest to the database. So, the turnaround time between the application and database is really low, so, the latency is really low. Also, on the other hand...

Karsten: Sorry to interrupt, that was the early HANA paradigm to get the algorithm to the data and set the data to the algorithm, right?

Yordan: Yes, exactly, exactly that paradigm. And, on the other hand, to have this HANA in-memory database. So, technologies or approaches to solve for certain business cases and combined together, they had this really nice use case. So, as you said, to put the algorithms closer to the database instead of the other way around. So, back then, it was quite a thing for SAP, and it faced a lot of adoption. And once we started with the SAP cloud platform back then what we had is this large SAP customer that adopted this approach, this paradigm, and of course, many customers in the cloud as well. So, they started using that. And it turned out that, after several years, that approach was not that good because on the one hand, it was a proprietary thing, so, it was very hard for partners and customers to actually contribute to guide the direction in which it would be going. And so, it became a pain. And also, at the same time, all of these guys were heavily invested in XS applications, and they had like hundreds and thousands of lines of code, and they spent millions for the applications for the support and so on. And what SAP is famous for is that companies' most critical scenarios actually run on SAP software, and that was the case for the XS application as well.

Yordan: So, it was certainly important for customers and partners to keep the XS development model alive. And SAP determined that this neo cloud platform, neo environment needed to be stopped, and that we needed to adopt to the modern world, to the Kubernetes, to the Cloud-Foundry-based one. And since the beginning of that year, maybe late 2020, we started thinking about moving these customers to the newer environment, and XS is basically the tool to do so. What we have in common with the Dirigible XSK and the XS development model is that in-system development experience. So, it's very similar to what we have in ABAP, so, you just write, for example, in XS JS code. So, it's again a JavaScript code. It uses a set of predefined APIs that are starting with this `doorsign.request`. And then you can use the request to get, for example, the query parameters or whatsoever. So, with Dirigible and XSK we can provide the same experience and the migration happens to be very easy.

Karsten: Okay, so, that's around, as you also said in the beginning, to protect these millions of invests that have been made into that programming model, right? That's

mighty nice of us. Does that mean it's pretty popular? Like, I don't know. Do you know how many productive customer landscapes use XSK-based applications? Do you have an idea about numbers there?

Yordan: So, that's actually a very good question, because right now, as I say, we started the effort at the end of 2020, so we've spent about a year on it, and now we're at the beginning or maybe the middle of this first adoption program. So, it's actually for early adopters that want to test the approach. So, we selected few customers and we're trying to convey to them and to run their scenarios. So far, we have like three customers, and they're huge, that have tested XSK. The feedback is quite positive, and I think that we scheduled a productive release or generally available release for the middle of 2021. So, after that, what I am hoping to see is this huge, huge adoption. So, actually there like thousands and thousands of customers that are using that, and we just need to get into the right place of maturity. So, not have as many bugs, you know? So, it's an open-source project in the end, and we want to provide the enterprise features that SAP is famous for. At TechEd, Jan Schaffner, he's part of the higher management, promised to provide enterprise-grade capabilities for the XSK project. So, we are starting open source. We're putting all the best effort, all the know-how, all the best practices that we have inside SAP to do open source and then run it inside the SAP for our customers. So, on the one hand side, their investment is protected because if one day we say we'll stop XSK and XS development models, the code, the GitHub repositories there, everyone could take it. Even a partner can take it and build it. So, basically, we guarantee that it won't die.

Karsten: If that now sounded interesting for some people out there, where do they go to find more, Dragomir? Where does one start?

Dragomir: As I mentioned earlier, first of all, you can check our website for Eclipse Dirigible, dirigible.io. There we have a section with our documentation. We have also a huge number of blogs written and published throughout the years, and we have a sample section if you are looking for examples, and we also have a link to our APIs. And of course, as I mentioned, you can click the "Try it out!" button to start the trial instance and check it out there. And of course, there is a link to our repository where you can post issues, start discussions, submit feature requests and yeah, feel free to give us a star there, as well. We also have a YouTube channel called The Eclipse

Dirigible, where we have around 30 videos so far. And we also have another channel, but on openSAP. It's a learning SAP channel where we have four videos so far, but we are planning to add more throughout the next year, and we're also available on Twitter, @dirigible_io. And for XSK there is a separate website, it's xsk.io, and yeah, it's under development, but feel free to check it out.

Karsten: Okay, thanks, and I'm sure we'll have the links under the link to the podcast, to all the major channels that you just mentioned. Now that we know where people go to get started, we are close to coming to an end. If you had two or three main points that our listeners should take away from this today, what would they be? You can alternate, or however you want to do this in short.

Yordan: I'll try to start first. So, Eclipse Dirigible and XSK are both open source projects. You're welcome to join in and co-innovate with us. Eclipse Dirigible is mostly for property developers and local developers. At the same time, it provides in-system development capabilities. So, you just write your code, save it, and it is activated on the fly, so you don't have to compile, you don't have to deploy. It's up and running in the cloud. And for the XSK, we preserve the XS development model, and it's really important for SAP customers and partners to do so.

Karsten: Okay. I counted three points here. So, Dragomir, you have a maximum of one that you can add if you want.

Dragomir: I think Yordan summarized it pretty well. I don't have anything to add. Feel free to contribute and contact us.

Karsten: Great. All right. You heard that. Feel free to contribute! For now, I want to say thank you, Yordan, and thank you, Dragomir.

Dragomir: Thank you, too, Karsten.

Yordan: Thanks, Karsten.

Karsten: That's it for today, this was the Eclipse Dirigible episode of the Open Source Way. You can hear us with a new episode every last Wednesday of the month, and

you'll find episodes on all the regular channels from openSAP to Apple Podcasts, Spotify and so on. Thanks for listening in. Hope you will do so again. That's it for today and goodbye.