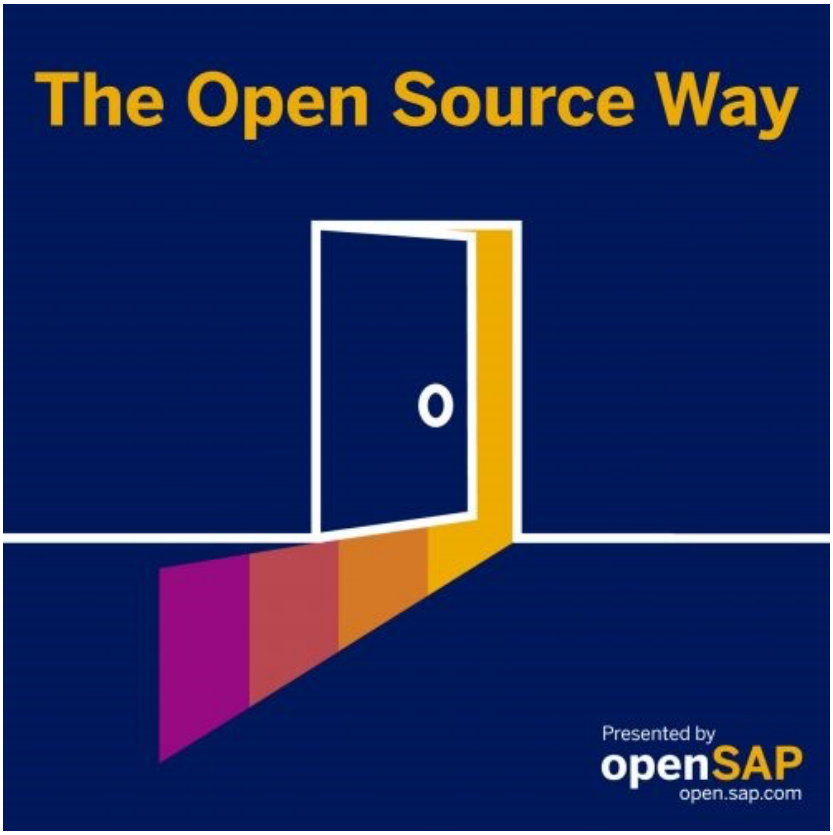
The Open Source Way

Episode 21: Open Source at Microsoft – ClearlyDefined and Open-Source Supply Chain Security



Transcript

Karsten: Welcome to the Open Source Way. This is our podcast series, SAP's podcast series about the difference that open source can be. And in each episode, we'll talk with experts about the open source way and about why they do it the open source way. I'm your host, Karsten Hohage, and in this episode, I'm going to talk to Nell Shamrell-Harrington from Microsoft about ClearlyDefined and possibly some more things. Also with us, to add even more things, is Sebastian Wolf from the SAP Open Source Program Office. You may remember him from a couple of previous episodes. Hi, Nell. Hi, Sebastian. Great to have you here.

Nell: Hi there. Thank you so much for having me. It's great to be here.

Sebastian: Hi there, happy to be here as well.

Karsten: Great. Sebastian, as I said, is from SAP's OSPO. I have introduced him a couple of times, I think. Nell Shamrell-Harrington, on the other hand, I have never introduced so far; she's a software engineer, a writer, and a speaker, and she happens to be a principal software engineer at Microsoft in the Azure office of the CTO, if I'm informed correctly. She's also a former lead engineer on the ClearlyDefined project and is also on the board of directors of the Rust Foundation. You may recall our episode about Rust from July 2021. If you don't, go back there and listen to it. And that means Nell is probably, at least to quite a few people out there in the open source world, a known name. But before we start, we plan to air this on May 25th. To me, that is Towel Day. So, all you hitchhikers out there be greeted. Now, does Towel Day mean anything to you or what does May 25th mean to you?

Nell: Well, May 25th happens to be my ninth wedding anniversary, so it means quite a bit to me. My wife and I will be celebrating together.

Karsten: Well, but probably not by going hitchhiking anywhere, right?

Nell: Probably not. It's not the safest thing to do.

Karsten: Okay. I imagine so. As I just said, you used to be or are just in transition away from being mainly responsible for ClearlyDefined. You will still be working on open source and with Microsoft in the future, still. But you will turn your attention more to the open source supply chain security. So, what should we start with, past or future?

Nell: Let's go ahead and talk a little bit about what ClearlyDefined is. ClearlyDefined is very focused on open source license compliance. That is what I started to work on when I first came to Microsoft and until recently, I led engineering on, while supply chain security is more focused on knowing all the open source dependencies of your software, verifying that they are what they say they are, that they come from where they say that they come from, and that they don't contain anything malicious. So, it's similar. It's mapping out your web of open source dependencies. But now I'm having more of a focus on security rather than license compliance.

Karsten: Okay. Did I get that right: ClearlyDefined is more about the compliance of licenses, like on the legal and business compliance side, while the open source supply chain security is more on the actual technical, on the code side. Are you more like the license or more the coding person, Nell?

Nell: Well, I am an engineer by trade rather than a lawyer, though, my wife, who I mentioned earlier, is a lawyer. But I am very much more focused on the code side. As the world of open source has changed and become more complex, I've taken a great interest in figuring out how to ensure that this open source software, you both use and create, is done so in a secure way.

Karsten: Let's maybe do it chronologically anyway, even though you just said you're more of a code person actually by trade or an engineer by trade. So, let's start with the past and go to future. What exactly, just for those who might listen, who don't know, is ClearlyDefined all about again?

Nell: ClearlyDefined is a central data store for all open source licenses across many different open source ecosystems. I think last time we checked, we had license definitions for over 14 million pieces of open source code. It crawls these ecosystems, harvests the license information from the package manager, the code repository, etc.,

verifies that the text of the license is consistent with the license that the software is supposed to be, and then it stores this information in a central place with a public API. So, anyone – Microsoft uses it, I know SAP does as well – can use this public API with their compliance systems to make sure that the licenses on any open source software they use are compliant with the way you intend to use them.

Karsten: All right. So, it's basically, if you want to call it like that, three tiers, it's the harvesters, it's the store and it's the API to make things accessible. Would that kind of describe it?

Nell: That's yeah, that's a good rough description.

Karsten: So now of that project, both SAP and Microsoft are members, right? Have there been many touch points in ClearlyDefined between Microsoft and SAP directly?

Nell: Quite a few. SAP employees have been really helpful contributors to the community. Sebastian, who's on the podcast with us, and Shane Tomlinson have contributed code. Brian Duran and others have contributed license curation to the data store, and SAP and Microsoft also did a webinar on ClearlyDefined together.

Karsten: Sebastian. I'm sure you can give the compliment back to Microsoft, right?

Sebastian: Of course, of course I can. So, it was really, really very nice to work with you, Nell. Also, actually, this webinar that we did together was actually the kick starter for me to contribute code back to ClearlyDefined. So, I didn't have the idea beforehand, so, I saw only, okay, well there's something that I'm currently working on that might be missing in ClearlyDefined. And then I had a look, took a look at the code, at the documentation, which is really nice, set it up on my local machine and got things going. So, that was really, really cool from my perspective.

Karsten: Sounds like open source is supposed to work, right? As you mentioned the webinar, we have to make sure because that we have the link with the information accompanying the podcast when we publish it. Next question, Nell. As large corporations, would you say Microsoft and SAP have similar roles in ClearlyDefined, or do they differ quite a bit?

Nell: I would say the way we use ClearlyDefined as consumers of it is very very similar. We build a lot of software that pulls in a lot of open source dependencies, and we have to know that the license on those dependencies is consistent with the way we want to use it. I imagine neither SAP nor Microsoft wants to pull in something where the license, let's say, prohibits the dependency from being used in a commercial distribution of something. We do produce a lot of commercial software. That's one of the reasons our companies exist. So, it is very similar in that we want to be good open source citizens and make sure we're using the open source we use in the correct way.

Karsten: Okay. So, you're saying as our use as consumers is similar, we probably have this at least similar focus of what we would like to see in ClearlyDefined and what we contribute. Is that right?

Nell: I believe so, yes.

Karsten: Okay. Okay. Now, for all I know, ClearlyDefined is a little bit subdivided or used to be, I'm not sure. I think mostly it's come down to licenses, but there are the sub-projects ClearlyDescribed, ClearlyLicensed and ClearlySecure. Are these all still around or...?

Nell: These sub-categories came about at the very beginning of the ClearlyDefined project, which was before I was working on it. And I know there has been some intent to explore all of these, but licensing has been the main focus of the ClearlyDefined project. Though as I mentioned before, in the past we've looked at ways of expanding it and we might see that expansion at some point in the future. ClearlySecure goes more into the direction of what I'll be doing in the future, which is focusing on open source supply chain security.

Karsten: All right. We'll come to that in a few minutes, I guess. Let's stick, for a moment longer, with ClearlyDefined. Does ClearlyDefined only use project owned, basically home grown, services or does it also call others from completely different, also community projects or even proprietary ones? I don't know.

Nell: Ah, we do use three services for harvesting license information. One is our own home grown, home coded one, but we also use two other open source license scanning

tools to get more data. One of those is scan code, which is a Python project, I believe, and then licensee which is written in Ruby. We use all three of those and then pull together all that information and see what's consistent, what's not. See if one of the scanners picked up something the other ones didn't, etc. So, we use two community projects and then one of our own community projects, if you will.

Karsten: Okay. Just one additional question on these. Are these basically sort of like specialized crawlers or how would you go and put it that way? Or what do they do?

Nell: They do crawl the source repositories. Yes. And they also crawl the package managers and pull license information from there. So, you could call them that, I believe.

Karsten: Okay. License specialized crawlers, basically, right? Okay. There seems to be another relationship that I only briefly touched when I was looking at the web. That's with SPDX. What's that? And how does ClearlyDefined use it, or what are the differences? I don't know what the relationship exactly is.

Nell: All right. SPDX is a few things, but one of the main things that ClearlyDefined uses them for is they have a list of identifiers for standard software licenses. So, one of the most well-known is Apache 2.0 or maybe the MIT license. ClearlyDefined only recognizes licenses that have an SPDX identifier. You can make up your own license however you want to. And there's been some controversy about that in the past few years. But ClearlyDefined only recognizes licenses that, one, have an SPDX identifier, two, we scan the license text in the repository or associated with the project and make sure that the license text in the project is the same as the license text that's associated with that identifier. So, when we say a piece of software's license is ClearlyDefined, we mean there is an identifier and the license text included with the software is the same as the license text you'll find when you go to the SPDX site.

Karsten: Okay. Well, probably before I'd even get to coding anything, being a non-coder, I'd be lost in this entire license topic after what you explained here. Anyway, let's just, because of that, not drill too much further into that, and maybe take it up to the larger perspective there for a second. Now, if we look 20 years back or something, open source was still, I almost like to say, something evil. At SAP, this attitude has changed,

at Microsoft and at many other large corporations this attitude has changed. What was it that changed Microsoft's attitude towards open source?

Nell: Well, it's no secret that in the past Microsoft had a tense relationship, I honestly would call it a toxic relationship, that's me speaking from my personal viewpoint, with open source. And any time I mention Microsoft and open source on Twitter or Reddit, a lot of people bring up a lot of quotes from 15 years ago. However, we have shown through our actions that we've transformed our relationship with both open source and the open source community. And frankly, it's better business to be building at least some of our software in the open and using open source software and contributing back to it. Now, we do use it extensively within Microsoft and we use ClearlyDefined to ensure that those licenses are consistent with the way we'll be using it. But we also know that it's not enough just to use open source. We need to be good citizens and contribute back to it. Now, part of this is through donating money to open source foundations like the Linux Foundation, Rust Foundation, and many others. But along with giving Microsoft's money, we also need to give our time and talent. Microsoft employees, some of the best engineers, some of the best product and program managers and community managers in the world. There is a lot to offer when it comes to expertise in building software, building communities. So, if you, any of your listeners are interested in learning more about Microsoft's relationship with open source, head on over to opensource.microsoft.com, and I'm sure that link will be included in the notes.

Karsten: Okay. So, Sebastian, how about at SAP? As I just said before, with SAP, it was quite similar, I don't know, I still remember when we wanted to use Python we had to buy some distributor package. And it was a big pain with the legal department and with the support colleagues who wanted to have their, I don't know, 40 page support amendment signed and everything before we were allowed to use anything. What's changed at SAP?

Sebastian: Yes, that's absolutely true. And I also remember these days really, really complicated. And yeah, everything comes down also to the fact that SAP is simply a company that makes its money on selling software and services; there's a reason for that. And yeah, SAP, we've gone through a pretty similar development of course. So, we may have started a little earlier with becoming, actually, a founding member of the Eclipse Foundation back in 2004. So, we were already active with community

engagements and similar things, but both the consumption. So, that was what you are referring to. But also, contributions to open source were really, really complicated back in these days. But things have improved considerably of course, and today we are using ten thousand of open source components in our products and services, and that's what Nell already said, we are not only consuming, but we are also contributing to prominent open source projects. And there are of course several examples for that. For example, the Java ecosystem with contributions to Open JDK and our own distribution with SAPMachine or the Kubernetes orchestrator Gardener, just to name a few. Right. And yeah, on the other side, especially in consuming open source projects, licenses still play an important role, and we still need to pay attention to the details. That's also the reason why we are using ClearlyDefined in our internal processes, right? And that's where our things come together. So, we classify our things, all the products and components we are using from an open source side in three different risk categories, low, medium, and high depending on potentially negative impact for our business. And we need proper tools to assess that. And yeah, that's where ClearlyDefined plays an important role. For our contributions these licenses are also really important, and there we also still need to pay close attention and that's where we are also good citizens, and therefore we also only apply low risk licenses under normal circumstances, like Apache 2.0 and MIT for our own open source projects. So, in the end we've become much more open towards open source, especially in the past few years. But of course, we need to balance risks and opportunities very well. So, yeah, that's basically it from our side.

Karsten: As you mentioned both sides now: On the one hand, we have to be sure about licenses when we're using them, and on the other hand, also when we are contributing to them. In the past, what was harder to overcome, and also in the present, what is still, for large companies like SAP and Microsoft, the bigger risk? Is it more the sharing code that we generate with an open source community, or is it more the shipping open source code with its different types of service level agreements and so on to customers? Maybe Sebastian first.

Sebastian: So, from my perspective, it would certainly depend on whom you ask and also on the respective circumstances at the moment, what they would answer, because especially in times of these security vulnerabilities, people would potentially answer that using and distributing software with open source components is really a challenge. But on the other side, also, many, many people would still say that contributing is the more

complicated thing, because SAP, as I mentioned, is still a company that makes its money by selling software services and similar things. So, many, many people in the company thought, and are still thinking, that we might give away critical intellectual property to competitors, you know, if we publish software as open source. So, on the other side, there's also a pretty interesting fact within the company and also in the whole SAP ecosystem, namely, that one of the key success drivers of SAP in the past, up till now and potentially for a long time in the future, is the ABAB ecosystem and everything around it, and that wouldn't be there, from my personal feeling, if the customers hadn't had access to the source code and if they hadn't been able to modify it, you know. So, it's technically not an open source license that the ABAP ecosystem is working in, but it's more or less some kind of shared source as you need a commercial license from SAP. But it clearly shows, from my perspective, the power of a more liberal approach towards source code, and that's also one foundation which we keep on using and leveraging to spread the word about the benefits of open source. And we clearly see the tangible results already. So, both the consumption and contribution statistics of the recent years show clearly that the idea of open source has gained a lot of traction within SAP. And we as Open Source Program Office are, of course, here, and responsible for keeping things going.

Karsten: Thanks, Sebastian. So, so much from SAP's side. Now, Nell, on Microsoft's side, was the bigger obstacle in the past, rather the sharing own IP or the using the stuff from the wild rebels that join up in communities out there on the net?

Nell: There historically was anxiety about both. There still little pockets of anxiety about both at Microsoft. How we address the anxiety people have of using a piece of open source software that's downloaded from the Internet is we have a lot of automated tools, some of which download the source code and rebuild it, and then we run scans on it. Some of them check any open source dependency someone pulls into a Microsoft project, verifies whether there are any known CVEs or other vulnerabilities associated with this version of the software. If there is, it breaks the build. And we also have that automated license checking compliance. So, we've built a lot of automated tools to help people feel more secure in using open source software and be more secure in using open source software. As far as intellectual property, there is a question I get often from people that are very new to open source, and that is "Well, if we put some of Microsoft Code up on GitHub, what's to stop failing a Microsoft competitor from taking that code

and using it and making money from it?" And my answer to that is that Microsoft does not open source all of its code. It open sources some of it; a lot of the critical intellectual property is still kept closed source. I'm sure there are people out there who think we should open source everything, and there are probably fewer, but some people who think we should Inner Source everything. It has to be a mix of both. What we have found through open sourcing things like Visual Studio Code, Azure SDK, software developer kits, is that the more people engage with our open source projects, modify them, contribute to them, connect with the community around them. I mean, if you look at it from purely a business perspective, the more likely they've engaged with us in the past, the more likely they are to consider things like Azure for their cloud computing needs in the future and for other things. So, yes, that's a kind of capitalistic business viewpoint of it. But Microsoft is a for profit company. That is one of the benefits. And also, you know, the goodwill benefits as well. And our software improves by other people engaging with us.

Karsten: I was just thinking, a couple of sentences before already, the rest is politics, and we don't want to get into that because you just mentioned the word ‘capitalistic’. I mean, some people think even supermarkets and all the goods in them should be open sourced and others think, no, they shouldn't. But that's politics, as I said.

Karsten: And that is not the point of this podcast. So, let's rather turn to your future before we run out of time here. Open source supply chain security. Can you give us the pitch and abstract about what that is?

Nell: Sure. So, I used to say everyone uses open source software now whether they realize it or not. Now I say everyone uses a very complex web of open source software dependencies, of dependencies, of dependencies, whether they realize it or not. Now, if one node in that really complex web is compromised, someone uploads something malicious to it or it disappears from the internet suddenly, I've had that happen, that can compromise every piece of software that depends on it. And it might not even be a direct dependency, but a dependency of a dependency. So open source supply chain security is a way of ensuring the security and integrity of every node in your dependency chain. And we are working on solutions for this. We are working on prototypes for it. It is not a solved problem by any means. We're still in the early stages, but it's an immensely important one to make progress on.

Karsten: Okay. So, would I, like, just from the concept, imagine that correctly, that one part of it is in the first place kind of mapping out the dependencies successfully between all these components that are not really yours, but they're open source from the wild rebels out there on the web, as I called them before. And the other part, then, is kind of like built in how do you call the places in the woods where there are no trees in English?

Nell: Clearings.

Karsten: To build a barrier against fire.

Nell: I don't know the word, but I'm following you. Yep.

Karsten: Sort of like this tree free highway through the forest. So, the fire doesn't get any more fuel along that stretch. No. Anyway, so, and then the next level would be thinking about kind of building these fire walls into your dependencies so that not everything gets corrupted if something is corrupted. Would that also be part of it or is that again, different?

Nell: It could be in the future, but at the moment it's focused on mapping out the web of dependencies and verifying those dependencies are what they say they are and came from where they said they are from. So, that's where we are right now at the very beginning. But I could see something like that in the future.

Karsten: Okay. But at first, it's more analyzing the interconnectedness and, I don't know, evaluating your risk that is connected to that?

Nell: And building tools to do that automatically, because there's no way to do that by hand anymore with hundreds of thousands of dependencies.

Karsten: Okay. I see. Now, with that, sorry that we touched that future of yours only briefly and dwelled mostly on the past, but it's always easier to talk about the past because that's already happened. And with that, we are reaching our famous final two questions. The first of those final two questions is always if people were interested and

wanted to learn more about ClearlyDefined or about open source supply chain security, where would you send them?

Nell: Sure. For ClearlyDefined I'd say head on over to docs.clearlydefined.io/getinvolved and I'm sure the link will be in the show notes. For open source supply chain security, we're still at the beginnings of it, but a project I'm currently working on in this space is called GitBOM. That's G-I-T-B-O-M. And you can head on over to gitbom.dev to learn more about it.

Karsten: And I bet Microsoft also has a prominent page talking about Microsoft open source engagement in general, right?

Nell: Opensource.microsoft.com, I believe, is the URL.

Karsten: And that will probably be the easiest to Google of all the URLs.

Nell: Absolutely.

Karsten: Okay, great. Thanks. And then finally, if you could make a wish, which two or three main points should our listeners take away from our talk today?

Nell: They would be that the dream of open source from the nineties and the early 2000s, that's the time I was starting to get into open source, that dream has largely been realized: Nearly every corporation, every government, every educational institution is using open source software and many of them are contributing back to that open source software. So, the dream came true. We won. And now the struggle is figuring out what comes next. And a lot of that involves understanding how do we use open source software in a responsible way? And that's why things like license compliance and open source supply chain security are so important.

Karsten: So, you're saying those people who I call the rebels out there on the net won, and if I remember correctly from Star Wars, it's good when the rebels win.

Nell: Generally, and then they had to figure out what comes next, too.

Karsten: Yeah, right. Okay. So, Sebastian, do you have anything to add to the takeaways?

Sebastian: No, I can only support that because this is when I started with open source. I think my first Linux installation that I did was in '98 or something like that. So, really an incredible journey we've been through over the last two decades, I would say. Yeah.

Karsten: Great. So, we're all agreed on that. Thank you very much, Nell, for being our guest today. And thank you, Sebastian, for joining us as well. We're saying bye at this point before I come to my final words. Thanks again. Bye.

Nell: Thank you so much. It was a pleasure being here today.

Sebastian: Thank you very much as well, once again. Pleasure to be here.

Karsten: All right, Sebastian, I'm sure we'll meet again in the office or in this podcast. And thank you all for listening to The Open Source Way. If you enjoyed this episode, please share it, and don't miss the next one. We publish every last Wednesday of the month and you'll find us on openSAP and all those places where you usually find your podcasts like, I don't know, Apple Podcasts, Spotify, whatever you prefer. Thanks again for listening and goodbye and have a happy rest of Towel Day. Bye.