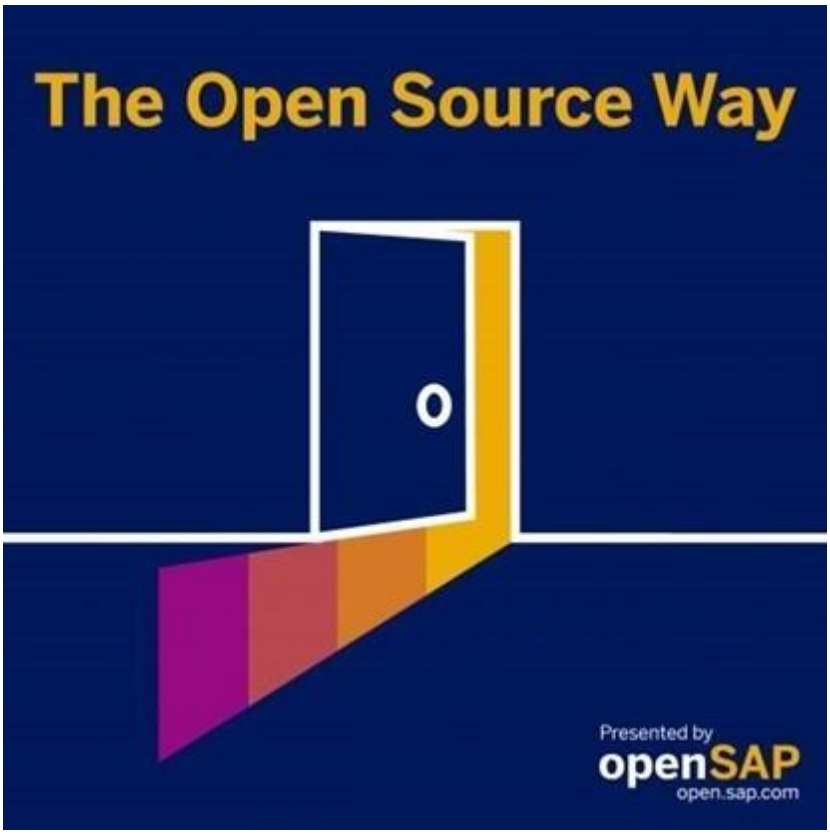
The Open Source Way

Episode 25: Linux Foundation – Building Trust in Supply Chains



Transcript

Karsten: Welcome to the Open Source Way. This is our podcast, SAP's podcast series about the difference that Open source can be. And in each episode, we'll talk with 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 Shane Coughlan about trust in open source supply chain in general, about OpenChain in particular, and probably some more stuff in that vicinity. Hi, Shane. Nice to have you here.

Shane: It is just fantastic to be here Karsten. I'm looking forward to our discussion. I think we're going to have a lot of fun.

Karsten: All right, let's look at who you are in the first place. Shane is a GM. Would you wish to explain that before I continue?

Shane: Of course. Of course. So, I'm the general manager of the OpenChain Project, and essentially, it's my job to help the community organize themselves, develop the material we present to the world. And naturally, if things go horribly wrong, it's my job to take the blame.

Karsten: Okay. So, GM, as expected, is general manager. You are also that in the Linux Foundation in general, right?

Shane: So, I will explain that my role is general manager of the OpenChain Project, and that's part of the Linux Foundation project portfolio. Our larger projects are run by executive directors or general managers, like myself.

Karsten: Okay. Let's look at some other points in your impressive CV there. You're also an advisor to the World Mobile Group, you're an assembly member of Open Forum Europe, you're an expert in communication, security, business development, and wow, things you have done. You built the largest open source governance community in the world through the OpenChain project. You spearheaded the licensing team - or you still are - that made Open Invention Network (OIN) the largest patent non-aggression community in history. Established the first global network for open source legal experts. Wow. You're founder of the First Law Journal on open source, the first law book

dedicated to open source. And currently, among other things, as you have just explained before, you lead the OpenChain project, right? Let's pick some more of that up. Governance, patents, legal aspects. That is apparently what you're all about. Are these like the non-functional aspects that add up to trust in open source or is there more?

Shane: I think that's a great way to put it. So ultimately, open source is a tool, and that tool happens to be software and we regulate this particular type of software tool with agreements, licenses. So ultimately, for open source to work, you need to have agreements, you need to have copyright agreements on the software code, you need to have patent agreements from the entities leveraging it. And to put it in a nutshell, we have to have governance methodologies that allow all these different stakeholders and competitors to collaborate on open source and to compete vigorously on products. So, my expertise is trying to navigate those waters, be it copyright, be it patents, or be it something else.

Karsten: Okay. But these legal aspects, copyright patent aspects, that's not all that there is when we want trust.

Shane: Trust in the supply chain. Yeah. So, you could say that we start from the legal side to allow us to begin moving in the supply chain. And then we have the supply chain positioned to run and immediately have the challenge "Okay, so it's positioned to run - how do you realize that potential?" And one of the key things is you have to have the parties trust each other for trade to occur. And that boils down to, first of all, getting the legal thing right, naturally, but having process management across the supply chain, that's predictable, understandable, and - if issues arise - remediation is easy. We can solve the issues together. It's particularly important to bear in mind that something global, like open source is going to involve parties which do not inherently trust each other. Therefore, we must create trust. We must have mechanisms that support it as quickly as possible. This all said I'm preaching to the choir here, given that your company touches 87% of the global supply chain - an impressive scope.

Karsten: Yeah, I think how we put it when we talk about trust in the open source supply chain there from an SAP side is, on the one hand, we need this license and copyright and patent and so on, trusting each other. We need the security trust in the software,

and we need the trust in what you have just mentioned, as there needs to be an agreement that if there are issues will fix them. That's basically the long term maintenance that has to be secured, right?

Shane: Right. And I think that touching on security, it's a great point. Often when we talk about open source and compliance, people immediately think license compliance and that is accurate. But security compliance is vital. Export control has always been vital. But now with the US-China tension, with the Russian tension, it's more important than ever. So really, I think we're talking about managing all these factors in play, with the end goal that you can deal with any supplier and customer company anywhere in the world with the minimal friction. And on the trusted side, you know, that means licensing, it means security, it means export control. It also means the simple fact that it's predictable. What you expect to get is what you get. And instead of people having to do that on their own, working out how to manage all of this, we're shepherding the global supply chain into shared process management. No reinventing of wheels necessary. And naturally, that's vital for onboarding more and more companies into a trusted supply chain.

Karsten: As you said, "shepherding", let's maybe get back to your personal role as a shepherd there. You if I have it down right, started with the Open Innovation Network, or maybe not started, but there you cover patents in particular and then you moved on to more general license compliance with the OpenChain project. I mean, I'm oversimplifying probably.

Shane: It's not inaccurate, but there was a bit before, so I'd call my career a ricochet across intellectual property. So, I started actually as the founding manager of Free Software Foundation Europe's legal department, and that NGO in Europe focused on issues around how open source or free software, as some people call it, could be leveraged effectively. And particularly in Europe, there were questions about would different jurisdictions, would people interpret the use of open source differently? Would licenses mean different things? And when I was doing that, in fact, communication across countries or even internal to countries was limited. I'll give you an example. There might be lawyers in Munich and in Frankfurt doing exactly the same job but not know each other. So, we started this network to bring these legal experts together in Europe. It quickly snowballed into a global activity. And that's, I think, where I put the

starting point of my career. After a few years of that, I shifted gears. I joined Open Invention Network, which was dealing with the patent problem as opposed to things like copyright. And at the time that was a hot topic. There was a lot of patent tension around open source. I'll just stress, not patent litigation, just tension, and concern. And Open Invention Network was the inherent solution where companies agree "Oh, we need this, so let's not fight with patents on this specifically.

Shane: We can still have our disagreements elsewhere." So, when I joined Open Invention Network, it was quite small. It was 59 companies in the cross-license, and we scaled by the time I finished, as global director of licensing, to 1950. And of course, they've continued without me and they're now well north of 3000 companies and other entities pledging patent non-aggression on open source. And then finally, I turned up in OpenChain. So, you could say I regarded the patent problem, if not solved, well managed and strong momentum. And the copyright issue was still bothering the supply chain. We still had a very simple problem, the touch points on bringing, let's say, a consumer electronics product to market, from silicon to end-packaged product arriving in your Amazon warehouse, would often involve 20 or more companies, and software would be moving between these companies to eventually coalesce into this product. And naturally, given that lots of different people and different legal entities were touching and contributing software, errors occurred. So, we've been trying to address that. And I mentioned before, we wanted to harmonize the way that people could solve these problems. And I'd like to stress something here. The global supply chain is terrifically big, but most of the parties in the supply chain are small and most of them have no resources.

Shane: Their margins are tiny, 1 to 2 to 3% for a lot of the companies in Taiwan or China. So, they actually don't have much money and certainly not much time to noodle on things like "how do we do really great licensing, so our end customers are happy?" So, one of our heaviest lifts was to distill the knowledge from the biggest companies in the world, who've spent the most time on governance. Squeeze it down, refine it so anyone can use that knowledge, and then try to deliver it in the supply chain to help people. So, you could say that the motivation here - transformation of the supply chain - was to actually empower even the smallest entities to get things more right. And that's not altruism, to be honest. The end goal is that as a big company, your procurement is just smooth, but it's a nice balance. And I think just like Open Invention Network, again

provided massive companies agreeing nonaggression so smaller companies could breathe a sigh of relief regarding open source and patents. What we do in OpenChain is hopefully make more and more companies breathe a sigh of relief. They know what processes to do. They can get the reference material and they can get tons of support. And that's the mental model.

Karsten: So, if I get you right, part of this was making the big ones understand that it's to their advantage, if also the small ones play by the rules, basically, or help them play by the rules by taking some of the effort out for them, right?

Shane: Yes. Yes. And of course, for a larger company, you need to sell the idea by saying, of course, this will lead to improved efficiency, resource wastage will be dramatically reduced. But only large companies really have the flexibility to invest in something that might take a couple of years horizon. So, it was a negotiation, but I would just stress that what was very cool, is that the large companies by and large have been enthusiastic, very enthusiastic to get this done. It's immediately recognized, yeah, this works. Let's, let's get it done.

Karsten: Well, I once saw a great example by one of my colleagues here. I mean, this is also a thing that's about standardization, basically, only of processes, not of APIs or whatsoever when we talk about legal matters.

Shane: Absolutely.

Karsten: And he pointed out what the overseas container did back in the seventies. It multiplied the trade volume, and that's always a great example to say if you standardize, everyone will win, right?

Shane: Yes. And the shipping containers, it's so surprising that it was one company trying to improve their deliveries and it snowballed into our global solution. For good or for worse, because they're not using metric system. But remarkable accomplishment.

Karsten: That's another thing here, about standardization. We still have some ways to go here when it comes to the units. Anyway... anyway, so that is a lot about legally trusting and trusting in each other's compliance processes, basically. How about the

next aspect of trust? How about security? How do we deal with trust in that in open source especially?

Shane: So, I think we can't point fingers at any companies or even any incident. We have to look in aggregate, what's happened in security for all software and what in particular seems to have caused issues in open source. Now, when you look at the large-scale picture, you'll find that some interesting metrics, basically show all software is just software. Open source is not inherently more secure or inherently less secure than proprietary software. And by the same token, using open source isn't inherently cheaper than using proprietary software. It's just your cost basis is in a different area. And of course, I will highlight one difference. Open source gives you access to incredible amounts of code, so that's a unique thing. Anyway, when it comes to security, essentially, you're looking at mathematically the same number of security problems wherever you go. With open source, our analysis is about where in particular does open source tend to trip up. And one of the key areas that trips up is not alien to proprietary software. It's wherever you have a software dependency which is maintained by a small group of people. Now in open source that might be volunteers. A while back, we had a huge security vulnerability called Heartbleed, where a cryptographic library was maintained, I think, by two people, both volunteers. And we were running the Internet on it, which wasn't ideal. And then the same thing happens in proprietary software when you have a small supplier company that might go bankrupt or whatever. But for open source, that was our key issue. If you depend on stuff, but it isn't adequately analyzed and funded from a security optic, that's a key failure point. And we had some other failure points that I think were interesting in open source. And one of them was simply that in the supply chain, if everyone can touch the code, then they might. And if someone touches the code and improves it. Quotation mark and quotation mark, they might introduce new security errors. So, you could have code entering the supply chain, which you think is fine, then someone noodles on the package and inadvertently opens something up. And I think that's where in the supply chain beyond saying, okay, small points of failure are serious in the supply chain, we're looking at touch points again. It's very similar to license compliance and mental model. Supply chain has a lot of companies, if people touch the code, something can and will go wrong at some point. So, you want to catch that with the processes and resolve it quickly.

Karsten: And here the institution or the organization that looks into the security supply chain, if we want to call it that way, would be the Open Source Security Foundation. Is that right?

Shane: Yes, it's one of. But it's certainly the project with the most momentum and terrific funding, wonderful leadership. So, it's I'd say, where we're coalescing as an industry. The American entities have quickly coalesced around it due to some political motivation as well. The White House had some executive order and additional guidance around improving security in open source and using SBOMS (software bill of materials) and so on. And in a nutshell, Open Source Security Foundation is looking at the areas of failure points and also providing some of the heavy lift, to allow smaller entities to have better security around open source. Faster, easier, and just like OpenChain is altruism for the purpose of benefit. Open Source Security Foundation is exactly the same. We need that supply chain fixed.

Karsten: We also joined that, right? The OSS Security Foundation, SAP did

Shane: Yes.

Karsten: Like kind of recently.

Shane: Yes.

Karsten: Did we have, or did you have much interaction with SAP because of that? Or anyway.

Shane: I think probably my interaction with SAP comes directly out of engagement with OpenChain. So, SAP is one of the entities that has taken an extensive adoption of OpenChain. So, instead of having simply a scoped program covering some products, SAP is whole entity conformant, and this is a great thing. Some other companies are as well, for instance, Arm and BlackBerry. But I was particularly excited about SAP because of your influence in the supply chain, and I think that opened so many doors for collaboration. So, I've been working with your open source people to do lots of these interviews, blog posts, and other things, and we're having tremendous fun.

Karsten: All right. Good to hear. Now, we've talked about legal compliance, trust in that. We've talked about security and how - again, similar concept - the big ones teach the small ones, if I may put it that way. So, we have looked at legal. Now, over time, how can we trust that it stays that way? Keyword "long-term maintenance."

Shane: Yeah. Actually, you've hit on probably the genuine challenge for the supply chain. People are often addressing things product by product, but ultimately, we're looking at evolutionary flows of software that - from experience in all software going back many decades, - we know the stuff hangs around and you put it in the new product and so on and so forth. So, any piece of software in play today is likely to keep appearing for 20, 30 years, and managing the software will not change. Our consistency in management, our ability to have sustainable processes, maintain the processes, and educate people is critical. Now to address this, OpenChain has taken some steps on license compliance. We've made an ISO standard for license compliance, and this means that not only is it codified, not only is it easier to use in things like procurement contracts, but even hypothetically, let's say all the companies funding OpenChain decided to fund something else, and the project just went into maintenance mode. It has this tremendous ecosystem with all of the various ISO certifiers, with the large companies leveraging it for continuity. And as an ISO standard, it's never going away.

Karsten: So, when you say "has this large community" would you say that a long-term maintenance is almost more secured in the open source world - at least once a project has become large - than it would be in the proprietary software world?

Shane: Absolutely. Because ultimately, you're looking at multiple points of failure instead of single points of failure. And with something like the OpenChain project, I'll just give you an example of our scale. We've got local language work groups in China, Japan, Korea, Taiwan, India, Germany, and UK, and these have hundreds of participants. I think the biggest one right now is China at 231 participants from roughly 200 companies. So, there's a lot of people with skin in the game and not just using the standard, but these are people actively contributing to the material around it and advocating it. So, you're going to have, if just from our work groups, you're going to have over 1000 points of failure to knock that section of this ISO standard. Then there's partner community, which is the vendors. For instance, PwC is an OpenChain certifier. The Chinese government via the Ministry of Trade's sub-entity "ICT" is an OpenChain

certifier and we have a lot of partners. I think we have about 80 or so now. Again, they have their own momentum, and they have a commercial motive to keep promoting this ISO standard. So, the points of failure are dramatically reduced. The continuity compared to a single entity is incomparable.

Karsten: So, these are the things basically covered by also some other projects - not only OpenChain: the project health, project liveliness, and so on. But I noticed that you come pretty much back to OpenChain with a lot of questions. Is that because it's your current baby or is it because an OpenChain it all comes together the licensed security and whatsoever?

Shane: I would suggest it's a bit of both, but mostly it's because things came together and when we started OpenChain, our vision was trust in the supply chain. And we started on licensed compliance, we made an ISO standard on that because that was the fire to put out. But our work groups around the world, like Japan, is currently operating eight sub-work groups, doing everything from software bill of materials through to open source program offices. So, I think OpenChain has snowballed, and we work closely with the other projects. We don't overlap on their domain, but I think we've snowballed into the high-level management of open source in general and we try to feed that to the projects that are more granular. So, I just think that this might be the culmination of about 17 years of noodling on how do we increase the high-level process management of open source.

Karsten: And if you say high-level management, that would also match to the fact, I guess, that most of the important players in these projects nowadays are really large organizations, right? So, the lone hacker as a contributor still exists but is not the main player in projects like this. Right?

Shane: Right. I don't have the very latest stats on this, but a couple of years ago, for instance, over 80% of the code in the Linux kernel, and that's over all the code. So, we're dealing with, I guess, 20 years of development, but the majority of the code, by far, is corporate intellectual property. And the volume of contributions today, again skews very heavily towards corporate intellectual property. Now, there could be a knee-jerk reaction "that's bad" and it's not. It means that there's billions of dollars of investment. And this changed the metric, I think, on how do we think about this stuff for

the future. And given that something like Linux is in everything from your phone to your air conditioner to your car, we have a lot of people with a lot of experience really working on that long-term continuity. And I'll just give you a very quick example, that I quite like. So, dealing with consumer electronics, the life cycle to market is about six months of good sales and then it tails off, and then you have the next generation one year later. Dealing with automotive, you have 5 to 7 years of product to market, you have ten years of support. But then, you're dealing with something like the Shinkansen train, 30 years. Shinkansen is the Japanese bullet train, and then you're dealing with a nuclear reactor, 50 to 70 years and all these different types are together. And particularly when you look at infrastructure companies like Hitachi or Toshiba, they've got this enormous half-century or more support cycles, for the code they're deploying right now. So, I think that really helped everything in the open-source supply chain to have those optics.

Karsten: Okay. So that means you have already touched the point I would have asked you about next let's call it real-world or tangible assets, projects like nuclear power plants, trains, and so on. Now, 20 years ago, when probably or 25 years ago, I don't know when probably the main contributors were still the loan hackers to open source projects; this kind of things nuclear power plant best example; would probably have called for very closed software.

Shane: Yes.

Karsten: What changed?

Shane: Right. No, I mean it's a wonderful question because, on the surface, it sounds insane to have code that everyone everywhere with no particular restriction is making and then put it in a nuclear power plant. I think what changed, was that sometimes our surface perception is completely wrong about the deployment and the actuality. So, in open source, everyone can contribute, but everyone has optics into what's there. And this is the idea that was floated early on, around 1998 that, you know, many eyes remove all the bugs. Now, that's not true, but it really helps. And when you're dealing with a nuclear reactor, by way of example, your proprietary software is highly, highly reviewed and it's very good, but it's not reviewed by this many people. And one of the differences here is that, not only one company is making a nuclear reactor, there's a

bunch of them, and they all want to use the same code. So suddenly, these domain experts are making sure that the code is viable, even though they're about to try to outbid each other in the marketplace. Now, I think personally, the real tipping point, was when the defense industry decided open source was the most viable way to get really difficult stuff done. And if I may make one call out, I choose the NSA, the US National Security Agency, as probably the best tipping point. They looked at Linux and they decided this is a good basis for the operating system in highly important, like Department of Defence Intelligence Agency operations. And they took Linux, they use their experts to make a security-enhanced version, called SC Linux, and then they gave it to everyone. And I think when something like the NSA, which is quite private, normally, you know, that's why there's a joke it stands for "no such agency". When they take something like Linux, they harden it, and they say this is what we expect to see running and critical infrastructure in defense. That changed it.

Karsten: Are we still in the area of things we may talk about? Or is somebody coming knocking on our doors tomorrow or tonight?

Shane: Ha! No, no, no, no. They have their own public websites on this and so on. So, you know, essentially, we had a unique situation where one of the most private bits of defense decided to be very public about something. And again, it was not altruism, it was the fact that to get, let's say the US defense industry working in that domain, you're dealing with, I believe it's over 10,000 companies in the US alone, so you can't go door to door selling the idea. You need to do something that spreads it, and open source was ideal.

Karsten: NSA sounds like a pretty good argument on the security side there.

Shane: It's not a fly by night start up.

Karsten: I sometimes think is it a little bit comparable also to the blockchain principle. If everyone has part of the key, then it's probably more secure than if one has the key.

Shane: Yeah. No, I think that's right. And in fact, blockchain is an outcome of open source, really. I mean, when the white paper came out and so on, naturally it was the open source lot that started with it. I mean, my colleagues were mining bitcoin on their

desktops in between work. And fundamentally, blockchain was an attempt to codify some of the benefits of mass distribution of trust. Now, I would argue that blockchain has some tremendous uses as an immutable ledger, but in the end, blockchain is basically a database you can only write to once. And it has its uses, but it's not a panacea for changing finance or changing how people interact. But it came from open source. The mental model was shared trust. And on top of that shared trust, there was a political manifesto and political concept.

Karsten: I didn't want to get into blockchain in depth there, and with all the energy consumption and whatever you can talk about when it comes to blockchain now, it was only about the concept of the comparability to the if it's not owned by one alone, basically. Now, we have come to a point, when I look at our running clock, that means I will start with my prefinal leading up to my very final question there. Prefinal question: where do people, be that lone hacker or be it the next big corporation IT guys, go to get started with OpenChain? Probably some pretty obvious links.

Shane: Right. And the traditional answer is to say go to www.openchainproject.org. But I wouldn't say that's always the easiest way to onboard something like this. I mean, ultimately, I'm in a very fortunate place that I've been steeped in this area for two decades. And our website is designed to be as easy as possible to onboard. But I would suggest, if you're listening to this in Germany, for example, or the UK, drop by one of our workgroup meetings, virtual or physical, meet your peers, find out their stories. This is truly about user companies using open source, helping user companies use open source. Because that economic growth is good for us all. If you wanted to take one of the slightly trite sayings: "a rising tide lifts all the boats." And we want to make sure that the boats don't have holes in the bottom.

Karsten: Okay.

Shane: So, have a look at our website and if at all possible, join into our work groups. Meet your friends and learn from them.

Karsten: And we will obviously also provide some of these links with the podcast. Now I have one suspicion. Did you, in your answer to links already include your key

takeaways? Some of the things just sounded like that. Or do you have some more key takeaways that our listeners should take away from this podcast?

Shane: Yeah. I think ultimately, we should have a model that our supply chain is complex and none of us are smart enough to fix the challenges we'll encounter. None of us are big enough to change the global supply chain. So, we have to work together. We have to work together to make things like license compliance work, security, compliance, export control, and plenty other. Working together is easier than ever now because we have projects like OpenChain, like open SSF, to help do some of the heavy lifting in building bridges. The key takeaway I would give you is that that's a really good thing. A second item I'd bring up is, the greatest advantage you can have in modeling management of open source is to understand that stuff like process management, like getting licensing done, like getting security done, isn't a cost center, it's an efficiency metric. What you want to do is have the least friction possible to go from prototyping, R&D, and to deployment. And that's our game. That's what we've been doing, and that's why you should be at this table. Ultimately, we want to save you resources, and that's again, not altruism. We want the supply chain to just really work very efficiently.

Karsten: Okay. If that second point doesn't convince anyone, I don't know what would. So, with that, thank you, Shane, it was great talking to you.

Shane: Oh it's... it's my pleasure. I loved being here. Thank you. And I hope I can come back someday because it's been a blast.

Karsten: I hope you will. Thank you. And that leads to my final words. And that's thank you to all for listening to the Open Source Way. Thank you again, Shane. If you all out there enjoyed this episode, please share it. Don't miss the next one. We usually publish every last Wednesday of the month. You'll find us on openSAP and in most places where you usually find your podcasts, Apple, Spotify, and the likes. Thanks again for listening and bye bye for now.