openSAP Invites, Episode 18

LEARN ABOUT EXTENSIBILITY CONCEPTS USING SAP EXTENSION SUITE

Transcript

Elisabeth Riemann: Welcome to openSAP Invites. I'm your host, Elisabeth Riemann, and in this episode, we're speaking with Product Manager, Karsten Strothmann, to learn all about the world of extensions and event-driven extensibility with SAP Extension Suite. Karsten explains when and how to extend the business core, highlights the different extension types available, and makes us realize the importance of staying focused and building for the future - and not just one for fair weather days. Karsten has a real talent for explaining technical concepts in simple terms. So, stay tuned to find out what a spider's web, a washing machine, and restaurant pagers have to do with extensions. When you remember these, you'll be well on your way to understanding and implementing extensibility concepts. Karsten Strothmann is a Product Manager and Evangelist for SAP Event Mesh at SAP SE in Walldorf, Germany. He has more than 22 years of experience in the software industry, 20 of which at SAP. Karsten's currently working on event-driven architectures and specifically extensions and integrations on SAP Business Technology Platform. Previously, Karsten worked on SAP Gateway, which can be seen as a synchronous technology for extending and integrating SAP backends. During his career, he has held various roles from product management to development to quality assurance and consulting. This has given Karsten an holistic view of both software creation and its usage. Karsten holds a Master's degree in Computer Science from the Technical University of Dortmund, Germany. Let's say hello. Hello, Karsten, welcome to openSAP Invites.

Karsten Strothmann: Hi Lizzie, thanks for having me.

Elisabeth Riemann: You're very welcome. Karsten, it's great to speak with you today to learn more about SAP Extension Suite and, in particular, focus on how we can extend the business core by building responsive, event-driven applications. Karsten, you're a respected evangelist for event-driven architectures and I think it's also fair to say that you're also well known for your creative use of metaphors to help customers better understand and remember technical concepts. Last year in the openSAP nutshell course "Introduction to SAP Cloud Platform Extension Suite" since rebranded, of

course, as SAP Extension Suite, you shared quite a few metaphors with us. Would you like to share one of your favorites with us now?

Karsten Strothmann: Yeah, I guess I can quickly get started with one of my favorites, the one that I typically share in respect to event-driven architecture is the spider's web example, and I'm not even sure whether we have shared that back then. You know, the thing is that technical topics are sometimes very, very hard to explain and sometimes they get too complicated for people to understand. And over time, you learn that real world examples fit very well to some of the technical solutions we provide. And in respect to the spider's web, you realize that the spider is kind of like an event consumer sitting in the middle of the spider's web and waiting for either a fly, a bug or leaf to end up in the spider's web. And if a leaf ends up in the spider's web, there's some vibration in the web. So, the spider gets informed on the significant change, what actually is an event, and then doesn't react to it because the spider over time has learned it's a leaf and unless it's a vegetarian spider, it's not going to react at all. And let's assume it's the fly. The spider would immediately react. And this is the same approach as a notification event. And you just learn that. And typically, I love travelling, as you know, Lizzie.

Elisabeth Riemann: I do.

Karsten Strothmann: You walk into these issues, you run into these issues all the time, and you just need to keep your eyes open, the spider's web is one of the examples. But like restaurants, all these places they offer solutions to real world problems and actually solutions to technical problems as well. So, I love that. I just love that approach.

Elisabeth Riemann: And I think it's so helpful for us to have those real-life examples, that are tangible for us all. And as you said as well, technical concepts can be really difficult to explain, really difficult to understand, and remember, too. And I think, you know, taking hobbies, travel, anything like that taken from real-life situations we can all identify with and if you can use those to explain something really quite complex, then that's wonderful. And I really do like the spider's web metaphor. That's brilliant. Thank you for sharing that with us to start with. Before we focus on event-driven architectures and what we mean by events and why they're important when building extension apps, can we start with the bigger picture? What is SAP Extension Suite exactly?

Karsten Strothmann: I will ignore the "exactly" for a second, because I think there's a very easy answer to that. In the end, it's all you need to extend SAP backends on the SAP Business Technology Platform. It goes for basically three pillars there, digital experience, meaning you extend different SAP backends by providing different means of access, like, for example, mobile access, like Web UIs, like speech. So different options there. Then there's the pillar that I typically focus on, that's the development efficiency pillar really developing on the Business Technology Platform to add something that is not available in the backend you want to extend. And then there's a process automation pillar, I focus on a little bit less. It's more about workflows and so on. So, building additional workflows, automate certain processes on the BTP to extend the backend. So, I think that's the easy answer. If you really want to go into detail there, then it's a suite of services and tools and it's going to deliver a consistent user experience and go via different digital touch points. It contains development tools, no-code tools, low-code tools, different mobile access channels, process automation workflows. And the idea of it is to, first of all, give you this entire menu of options and at the same time guide you through to make it easy. So, I prefer the easy explanation. So, it's everything you need to extend SAP backend systems.

Elisabeth Riemann: Perfect. Thank you. I'm going to ask you one additional question when it comes that you mentioned BTP, would you like to tell us what BTP stands for in the context of SAP Extension Suite?

Karsten Strothmann: Yes, sure, SAP BTP stands for SAP Business Technology Platform and is SAP's cloud-based platform for the Intelligent Enterprise. It does focus on three scenarios and these main scenarios are integration, meaning to connect all your systems seamlessly, no matter whether we're talking on premise systems or cloud systems, there. Data to value, that means you provide customers with a consolidated view of all their data assets and deriving value from those assets. And this is obviously the main scenario in the context of SAP Extension Suite. Extensibility, meaning how do you extend your SAP backends to adjust them to your specific customer needs and use cases?

Elisabeth Riemann: You've told us then in a nutshell, what SAP Extension Suite is giving you everything you need to extend. Can you explain the business challenge a

little bit to us now? What are some of the problems and complexities typically associated with extending core business systems? And how does SAP Extension Suite address these?

Karsten Strothmann: Let me start with why would you want to extend in the first place in the end, almost no customer is going to keep an SAP system as it is. They always adjust the systems, kind of like if you buy a new car, I mean, you're probably not going to take what the what the manufacturer offers you. You would customize it in a lot of ways. And that's what customers do as well. They really go and want some adjustments. They are their business processes are a little bit different. So almost all SAP backend systems get adjusted. Typically, there are means in the backend systems to allow for that. Sometimes those means are not sufficient. The example I was typically using is the trailer scenario. Actually, it's a scenario that I actually experienced the year before, around Christmas. I had been selling a washing machine and the guy who wanted to pick it up, try to put it into his automotive core actually, so into his car, and the washing machine didn't fit. So, he was putting that on the trailer, transporting it home on the trailer and worked perfectly. And that's really the idea there are some business cases, some use cases for which the means that the SAP backends provide for enhancing, for extending are not sufficient. The Extension Suite tries to facilitate all of that.

Elisabeth Riemann: And I love that example that you use as well with the trailer and taken again from real life; you are selling your washing machine. The person who is buying the washing machine can get it to fit in the car. So, it's all about enhancing what you have and really making it into an extension.

Karsten Strothmann: Exactly.

Elisabeth Riemann: Thank you, Karsten, and obviously, depending on the business scenario that we look at, different kind of extension approaches are necessary. So, what types of extensions are there and when should we focus on which particular one? Can you talk us through the different types available to us?

Karsten Strothmann: The one that you would typically go for first is in-app extensibility, so you would just do a few changes, adjustments and the SAP backend systems that

could even be done by key users. So, they are not going to change anything really in respect to the to the underlying configurations and so on. But on a higher level, you can adjust, for example, languages and things like that. That would be one example. You know, it doesn't really have an effect on your on your backend system, but you have configured it to fit your needs. That and there are further steps that you could take. Let's quickly go to the car comparison here. In the old days, that would, for example, have been a radio that gets upgraded to a CD player. So, you just it's your car. So, you're just exchanging the radio to the CD player or you would put on winter tires or nowadays that would just be switching on additional features in your car using software. So that is basically in-app extensibility enhancement that really don't have any major impact when upgrading and so on. Then then there is the classic extensibility. So, this is a concept most people that have been working in the SAP area for a longer period of time and have been working with ABAB. We're talking about that coding here. So, you would really code in the customer namespace. And obviously these are bigger changes and these changes are not available in every single backend system. For example, there are cloud systems where you cannot go for classic extensibility and in the cloud, that's not the typical approach as well. And the cloud, it's typically side by side extensibility. If you want to go for additional code. Let me first give you an example in respect to classic extensibility. To me, that would be like a roof rack that you put on top of your car. So, it's a bigger change. You can still remove it at some point in time. So, you're not changing the car completely. Nevertheless, it's something bigger. It's not just a just in-app like putting the backseat flat to transport the washing machine. If your car is big enough, it is really a bigger change and takes more effort. And then, as said, primarily in the cloud on the is side-by-side extensibility. That is the trailer example. So, you would build additional steps for your business processes. It could be coding, no coding could be obviously workflow steps on the business technology platform there.

Elisabeth Riemann: Thank you for that summary of the different types that we had in-app extensibility, and then you focused on classic extensibility and then for the cloud side-by-side extensibility, where we get your metaphor again with the car and the trailer, which I love, by the way. So Karsten, if we can stay with side-by-side extensibility a little bit longer, can you talk us through some of the key advantages there?

Karsten Strothmann: Yes, there are actually three areas, business advantages, advantages in respect to technology and advantages in respect to operations. Let's start

with the business advantages. First of all, not everybody has access to a backend system. And a lot of cases you don't want externals to access us for a system, for example. So, if you build a side-by-side extension, you could address new user groups, for example, call center personnel. Now, that is not part of your company that you don't really want to access your backend system or external companies that would perform certain business steps for you. We have some examples that you can actually implement. And one of those examples is a business partner approval by an external company, for example. Another major advantage is innovation since you're not working or extending your core solution in your backend system. There's obviously a lot more flexibility in respect to what you can do for innovation. That's really good. Adding additional step and optimization is quite good as well, because for different countries you could add additional steps. I believe at some point in time when people have enough experience with this kind of architecture, they will bring together data from different backend systems. So, you take data from SuccessFactors, you take data from ERP and you bring that together in an extension application. You could actually even take data from SAP external sources and just mix that up and build a really nice extension application in respect to operations. You test in one place, you extend in one place, you have to just operate that central place for a number of extension applications. You remove always-on. There are means, if you trigger your extension by events, meaning if there's a change in the backend, it automatically starts up your extension. There are means of just being able to start up the this extension application when needed. And in respect to technology, the good thing is that you can pick the technology for your top of choice. If you don't have people that know ABAP well to build on that coding in your backend system, you could basically go and have them code in Node.js or Java, a lot of the younger people, but they have less about experience, more experience than other programming languages. So, it's the technology of your choice. And you could really build up that de-central know how they are as well and extend different backend systems with that. So same technology for extending ECC and S/4, for example.

Elisabeth Riemann: The benefits really are huge, aren't they? It's complete flexibility and then the aspect you mentioned there as well about the alternative to always being on right, the responsive side of the extension applications that you talk about when you bring events into the mix, then too.

Karsten Strothmann: So, lots of advantages. The one that we always push forward at events, it's keeping the core lean. So, upgrading your backend gets a lot easier if there are no extensions really in there that's that you have in the Z code space or where you would have really modified things. You keep all of that on the Business Technology Platform and your core is kept lean, testing is made easier, upgrading, and so on.

Elisabeth Riemann: That's key, as you said, isn't it, keeping the core lean and focused and really keeping those separate interests then there are as well. So, we've looked at the advantages there of which they are immense. Can you talk us through the best practices for building extensions? This was included in the course. And you shared an example here with the kid's room challenge. And I think this really helps us remember when to take which approach. Can you talk us through that?

Karsten Strothmann: I believe you, you should go for event-driven architectures only when they make sense and when it's really needed, because the first concept to check out is in-app extensibility. So, use the means that the different backends provide, because that's the easiest way. If that is not that's not sufficient, then I believe you have to plan ahead. And this is really what the what the kids room challenge means. In the end, if you if you think about your kids, they are born, they probably get their own room and then you start buying furniture there and you plan probably for the first two, three years and those kids grow. And at some point, in time you have to start exchanging furniture so your daughter or son will probably need a bigger bed or there might be a different chair that is needed and so on. And there are different approaches to that. So, you could reuse what is there and just exchange selected pieces. You could renovate what is there by adjusting things in certain ways or you could renew everything. And I believe you need to go for a certain mix there to make all of that work out properly. You don't want to renew everything all the time because that gets really expensive. Reuse is certainly going to end once your kid's legs hang out of the bed too far, probably. And yeah, renovating

Elisabeth Riemann: Outgrown.

Karsten Strothmann: Is only possible to a certain point as well. Renovating in that case means you keep most of the stuff and just invest in very important pieces. The point here is really think ahead and look at what makes sense. In some cases, go for

extension applications. That is probably the renovation approach. Plan ahead and keep these changes in mind. I mean, your landscape is not going to stay unchanged for years and years and years, specifically, not in these times with AI, with major analytic topics just around the corner. So, yeah, and I think the side-by-side extension approach, the event-driven extension approach is one way forward here. There's another bigger topic. Keep the backend in the lead because this is where you have this is where you have the data, and this is what the source of truth is. And in the end, it comes down to a data versus events topic. I still believe that the that the backend should be in the lead. There are different schools. So, in respect to event-driven architectures that basically say there's a certain shift towards event, a lot more of the truth is being kept in events. I'm not sure whether this is really going to fully ever kick off because it does have a few downfalls. So, from my perspective, and it's obvious, I mean, most SAP customers have lots and lots and lots of backend and storing all of that information in events and keep it flowing around would be very, very difficult. So, I still stick to that. And I believe it's the easiest approach as well because you want to ensure that you that your data is right, that your data is up to date. Starting with event-driven architecture, is a good point here. I highly recommend to start small and then grow because this really is a different beast when going for event-driven approaches. You want to, you want to learn, and you want to play around with this different architecture type first, because in the end you have a lot of advantages. If things go wrong though, it can be very challenging to track what has been going on. Which brings me to the nice little sunny weather example I have been using. So, don't build just for the sunny weather, because it can be hard to track what had been what had been going on. Like it's a normal thing in distributed environment obviously expect bad weather. So always make sure that you have some, yeah, I guess I would put it checks and balances in there so that that things cannot, cannot, cannot go too wrong. And that that there's a certain stop and that you get informed on when, when, when things go wrong, because that's one of the things that is challenging in in respect to event-driven architectures. Everything just flows. You don't have to interfere too much. It works great unless something goes wrong, and you just want to plan for exactly that moment. There's another thing. And that's the last point I quickly want to bring up. Obviously, you can only build extension side-by-side extensions when there are extension points in the backend system. So, for APIs, it usually looks very good in the backend systems. For events you need to check in in the API Business Hub, so api.sap.com, you can look up available events and currently for some of the backends, it's look, it looks really good. So, you can get almost every event

that you want for some of the backend, it's more challenging. So, there's only a limited number of events there currently. And that's one thing to check on before building events and then for building extensions.

Elisabeth Riemann: We keep mentioning events, and I think it's important perhaps that we explain what are events exactly. We talk about events, events sources. Maybe you can put that together in context for us and explain in a nutshell what we need to know there.

Karsten Strothmann: So, the different definition is very short, actually, events are significant changes, meaning a business partner getting updated in an SAP backend system. So, you just have, you change the address of the business partner and that is an event. So, a significant change that impacts your basic business, basically. And the advantage is that these events allow real-time approaches to react to that change. And if I really want to go into detail, then the event is a significant change in the backend system. You turn that into an event description, meaning there's a context, let's say, a time when did this event occur, what type and so on. And then there's the event data. So, there's typically two types of event data here, depending on whether that's a notification event that just informs you of a change and just keep the minimal data, meaning business partner IDs one, two, three has just been changed. And then there are data events. They hold the full data. So, they are different event types. Those events, these descriptions get turned into a message. They are handed over to an event broker. In our case, it's called SAP Event Mesh. And then those events are transported across your landscape to basically everybody interested. And everybody that has subscribed to receiving those events can then execute on those events and take some action. This is basically the idea. Why is this so important? Because it's asynchronous. That's another example. If you think about the police, they stop you and they check on your driver's license every now and then. And while they check on your driver's license, you are blocked. So, it's basically a synchronous approach. And once the policeman has given you the driver's license back and told you that everything is OK. You can keep driving again. But this typically takes a minute or two that that you are blocked from driving. If you go too fast, you have speed cams and these speed cams take your picture if you go over a certain speed limit. It's not nice to receive a ticket there either. Nevertheless, you can keep driving, it's asynchronous. It takes the picture. The system checks whether you're too fast or not. If you have been going too fast, you will receive a

notification sometime later. And this is basically the approach. You are not blocked. You can continue with your with your work. This holds true for the sender of the event, so the backend system just sent the event and it's not going to do anything, anything else there. And the receiver will get the notification, or we'll get the data event on that there has been a change, or a new business partner created, for example, and then can keep working. And before, neither the sender nor the receiver of the message. In this case, the event source, and the event on consumer, neither of them is blocked. And that's the beauty there.

Elisabeth Riemann: And it really safeguards performance, then it's non-blocking asynchronous communication.

Karsten Strothmann: Exactly, it's non-blocking asynchronous communication. It scales extremely well. So, these are benefits that it basically inherits from the underlying messaging layer. It's scales extremely way. Well, it's non-blocking. It allows for completely different approaches. And it's really real-time. Let us another major, major, major thing.

Elisabeth Riemann: So that's the service SAP Event Mesh that you explained to us there with the event broker and.

Karsten Strothmann: Yes, the event broker we're using is SAP Event Mesh and then there is that there are the event sources. Those are typically the backend system. It could be any microservice as well. It could be some SAP external event source. And then there are different event consumers in this picture as well.

Elisabeth Riemann: Karsten, can you walk us through how to build an extension, what services and technologies do we need? What's the end-to-end process look like?

Karsten Strothmann: There are two end-to-end processes. There is a quick process which I would use for doing PoCs or trying things out, that would really be to configure the backend system, to connect it to the Business Technology Platform. Actually, the order would be you first connect the backend system to the Business Technology Platform. There are tools for that, and they support you quite some of these days. Then I would configure the backend system to expose the events. Then I would get go ahead,

configure SAP Event Mesh and check whether the events arrive in a concept called Queues on an Event Mesh. And then I would build an extension application against SAP Event Mesh. So, it's quite quick if you want to try it out there are missions, they are on the on the Discovery Center, which provides coding. Actually, you can find it on GitHub as well. That provides coding and end-to-end descriptions how to do that. There's a more realistic approach to this, though, because typically what you do, you develop for a customer and you want to have some feedback cycle there. So, what I would do there, is I would not go for the for the backend first. I would start by building the extension application first using mock APIs and simulating the events first. That would be my approach. Then I would build the extension application and I would show it to the customer. That is actually the approach CAP is typically taking. So they build the app first, make it make it work, and then they go and build up the infrastructure there in respect to connecting the Business Technology Platform to the backend system to configure the event exposure in the backend system and then showing it to the customer again, you would have a full working app. The advantage really is that you get customer feedback.

Elisabeth Riemann: And talking about customers, can you share some customer business scenarios and use cases with us?

Karsten Strothmann: I would like to, again, differentiate a little bit here between the Extension Suite scenarios and event-driven scenarios, because in the end Extension Suite scenarios are more mainstream these days, meaning you could use API as well and that is most of us probably already have experiences with APIs and customers have been using those for a long time, really to build extension scenarios. So, one example is the Covid-19 app that that Germany built to get its citizens back to Germany at the beginning of the of the pandemic when everybody was who had been traveling was stuck all over the world. So, they built this this application in respect to organizing how to get people back and people were able to register there and so on. So that worked very well. There's another example that I like that is more classic as well. A chemical company, they were taking big orders there. So, it's a really large company and they took their orders still by phone. So, they have built an extension to their backend system. I think it was an ERP to build a portal for taking these orders. And obviously they are quite that quite something. That's quite an improvement there in respect to the to the business process. So that is more like really traditional approaches

in respect to doing extensions. If you look at event-driven architectures, there. My favorite scenario is still the TechEd scenario from two years ago where there was this car that you ordered. And this is I like it so much because first of all, it's something that I usually complain about You order a car, then four months later they call you. The car is ready. You can pick it up. I believe they have some upselling opportunities there. So, what we did back then is, when the car was getting painted, there was a certain status change to a business object in the backend system, and that did trigger an event. The picture was taken of the car during that process. And there was there was some actually back then it was a it was a serverless function that got started up. And it collected information from different backend systems combined with this information and informed the customer about the car, that his car just got painted and it included the picture in there. And that's just that's just absolutely awesome, because, first of all, I see the first picture of my new car, and that is certainly going to make me happy.

Elisabeth Riemann: It's exciting.

Karsten Strothmann: Yeah, it is exciting, and it keeps the customer engaged. And on top of that, there was there were a few upselling opportunities at the bottom of this nice little email. Hey, don't you want to order the bigger tires on top, you can still do that or something like that. And this is exactly the kind of scenario that that you should go for. I mean, it's a differentiating scenario. It will never be part of a of a standard SAP backend system. Let's be realistic.

Elisabeth Riemann: It shows that the power of event driven. Right. That you said you have a business object in the backend system, and it can really trigger an event and that updates the application as well. So, it sends an update to the recipient and then you can really see your car in that example. And I think that's something that we can all well imagine and really see the benefits there, too, which is fantastic.

Karsten Strothmann: I'm not even sure whether you would try a scenario like that, it's very innovative, and if you would build that in your in your backhand, you know, the cost would probably be too high. And you never know whether customers would really go for it. So, you can just build it on the BTP. And if it fails, it fails. It doesn't cost too much to build it. You can just switch it off and that's it. There's one more thing in respect to the scenario, why I like it so much. It does take data from different backends. And in respect

to real customer scenarios, most customers are not there yet, so they go for easier scenarios that typically just take advantage of the real-time advantages, the real-time opportunities of event-driven architectures and of the buffering. So, in a lot of cases, for example, it would be a project approval scenario. So, there's a new project created, and you need manager approval and automatically there's an event raised, and they answer to that. Or they have been they have been SuccessFactors HR scenarios in respect to learning solutions that the customers have built, again, strongly build on the on the intelligence services events and SuccessFactors that are available there. A few other smaller scenarios like Printer Management. So, when there's a printout happening, some event is triggered, and you take some action. So, customers are getting started. They are not there yet in respect to the really big scenarios. And actually, I told you before my best practices, this is exactly the way to go, start small and then grow from there.

Elisabeth Riemann: What exciting innovations are to come this year for the remainder of 2021?

Karsten Strothmann: Well, first of all, and I'm not excited about that, there has been a renaming from Enterprise Messaging to Event Mesh. I think it is, I am excited about the message behind, nevertheless, we've all gotten used to Enterprise Messaging. I still believe the renaming had to take place because we're moving away from messaging to eventing. I think we had not talked about this too much, but Event Mesh, Enterprise Messaging back then, had started out as a message broker. So, you were sending whatever kind of data to some receiver from one microservice to another and so on. And over time, this has with the focus on integrating and extending SAP backends strongly shifted towards event-driven architectures and towards sending events. So therefore, there is this name change to Event Mesh. And on top of that, it's really about building an event network on.

Elisabeth Riemann: And that's where Mesh comes into it, right?

Karsten Strothmann: Exactly. And this comes from both the event sources, meaning more and more backends being event-enabled, but as well as in respect to the internal setup of the of the event service, that we're basically offering, the Event Mesh service. So that is one thing that that that doesn't necessarily excite me in respect to the name

change. It does excite me in respect to the opportunity. So, the other thing that has been the main thing last year is really the event enablement of ERP, there has been this new this new add-on to ECC. I think it's called Event Enablement, SAP NetWeaver Event Enablement Add-On, something like that. And the beauty there, is that now all of those ECC systems can fire events and it's not, it's really an SDK for events, if you want to put it like that. So, it does offer so many opportunities and it's so close to what I would really would really like to like to have and would wish for in respect to creating all the events that you want would want. I think you can create thousands of different event types in ECC now. You can not only fire events, not only produce events on the ECC side, you can consume events as well. That's another major factor. And you can actually adjust to events to exactly the sizes that you want. You can send notification events very small, just containing the necessary and absolutely needed information. You can send big data events. It's completely customizable to your needs. That's a really exciting thing because it offers all the business opportunities. And let's be fully frank here, in some of the SAP backends, the number of events is still limited and you need to have a certain breakeven point to make it worth it to use events, as said, I'm really excited about the ECC Add-On because we we've had so many customers asking for event enablement for ERP.

Elisabeth Riemann: That's really good.

Karsten Strothmann: Yes, I'm really happy about that. One more thing, and this is something that sounds small, in fact, it's really big. I have been mentioning that you need to event enable the backend system and connect that to the Business Technology Platform as well to be able to obviously transport the events. And there the colleagues have been able to reduce the number of steps required to set these connections up and to do those configurations. They have been able to reduce this big time. So, when I heard this, I was smiling as well. And when I saw the numbers, I do like that a lot because I have to do that every now and then. And the less steps I have to go through, the better it is for the customers.

Elisabeth Riemann: Absolutely, I think we all feel like that is one big leap forward then that's really good. And Karsten, on openSAP Invites, we really like to highlight additional learning resources and assets. So, what do you recommend to learners who've

completed the introductory course, CP11? What should be our next steps? Where can we find further information?

Karsten Strothmann: Well, I think they are there are two things here. I think there are people that might want to get hands-on, those certainly should go to the Discovery Center. If they somehow got access to SAP backend systems, to the to the SAP BTP, there are a number of missions, ready to run projects, basically, and there are more every week, specifically in respect to event-driven architectures. Pick one of the easy missions, the smaller missions, and get started. On top of that there are learning resources as well. So that is certainly one way to go with a strong focus on event driven architecture. Nevertheless, there are missions in respect to the Extension Suite as well. So, so basic, more basic API based missions, workflow missions and so on. I would think I would check that out. The other thing that is, that is down the road, we're actually working on an online book on an ebyte on event-driven architectures that should be out sometime soon. So, if you're specifically interested in event-driven architectures, that would be a really good read, I think. Working hard to making it a really good one. And let's see what's going to be left once we once we finish the book. I would try to put all a lot of those examples in as well to make it as easy as possible to understand, while at the same time being really accurate on the definitions. And we will look at all the different backend systems as well and will try to provide hands-on information there. So that's the second thing. And yeah, if you want to look at the at the bigger picture, I would really go and look at the different Extension Suite resources there that that that are available. And that would be the standard SAP channels. I think. Discovery Center, again, there are two sections there, there's a Missions section, there's as well the Services section and this would be another thing I would highly recommend is all the services of the BTP listed that they are ordered by Extension Suite, Integration Suite, and so on. So, you could drill into details there.

Elisabeth Riemann: Karsten, to conclude today's episode, please, can you help us remember three main takeaways from today's discussions and metaphors are allowed.

Karsten Strothmann: Yes, I will stick to the metaphors, trailer and washing machine. Let's try to see that picture in your mind. That is actually the extension approach. Remember, in-app extensions is what to go for first. If that does not work and you have the right backend system, you can put on a roof rack, meaning you can go for classic

extensibility. And on top of that, there's a trailer that you can attach to your car and that would then be side-by-side extensions, which are for selected business cases and that allow you both optimizations and innovations. Then, and that was probably the one I want you to remember most. And this is an event-driven metaphor. That's a fair weather, bad weather example. So really plan for bad weather and things can go wrong. And with event-driven architectures, you have a lot of advantages. Nevertheless, you want to make sure that if things go wrong, you got things under control and you can track what has gone wrong. We've been talking about event-driven extensions for a major part of the sessions because this is one of my actually my main topics these days. And I do understand, though, that this is not the solution to everything. So, there was this one example, real world example when I was driving home from skiing with my with my son, he got hungry. We stopped by at the burger place on the way because he desperately wanted chicken nuggets.

Karsten´s Son: Papa, I want chicken nuggets now.

Karsten Strothmann: He did not stop asking for them, so I stopped at the first restaurant there and there was they were just taking API-based approach in the end, a synchronous approach. So, you had to get in line to order and then you had to wait to get your food. And there were 40 people waiting in line. So, I was like, no, I can't do that. So, I continue driving. And I stopped at the second place and they were doing things in a very, very smart way. So, they had an API-based approach next to a notification event approach, next to a data event approach, meaning you could either order your food and wait or you could order, and they were giving you a restaurant pager. Then you could sit down at the table and wait for the restaurant pager to make some noise to tell you that the food is ready. Then you would pick it up. And there was a there was a third option. They had basically the real-world analogy of a data event. So, they had a waiter serving you the food to your table and there was no line, it worked so smoothly. And this is the real learning there. Try to figure out what fits best, mix the technologies and approaches there.

Elisabeth Riemann: Fantastic. Karsten, thank you so much for being our guest today for sharing all those wonderful metaphors with us. You've so much knowledge of extensions and event-driven architecture. Thank you so much.

Karsten Strothmann: Thank you very much, Lizzie, for having had the chance to talk about it. I really enjoyed it.

Elisabeth Riemann: Thank you for listening to openSAP Invites. If you enjoyed this episode, please share, and leave a review, and don't miss your next invite. Subscribe now.

www.sap.com/contactsap