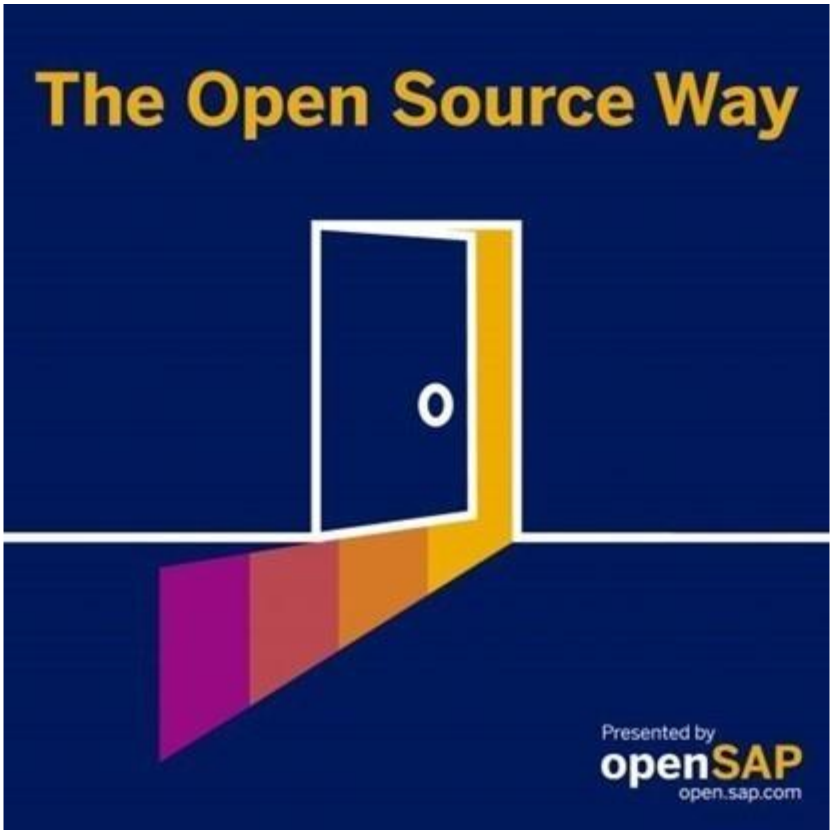
The Open Source Way

Episode 28: SAP Cloud Application Programming Model (CAP) – A step-by-step journey towards open source



Transcript

Karsten: Welcome to the Open Source Way. This is our podcast series, SAP's podcast series, that is about the difference that open source can be. And in each episode, we will talk with experts about open source and why they do it the open source way. I'm your host, Karsten Hohage, and in this episode, I'm going to talk to Christian Georgi and Daniel Hutzel from SAP's Cloud application programming model. Known maybe to some of you out there simply as CAP, C-A-P. About CAP - especially in this podcast - gradually going open source, that is. So, hi Daniel and hi Christian in the first place.

Christian: Hello. Hi, there.

Daniel: Hey, Karsten. Nice to see you.

Karsten: Nice to have you here. A few words about you two from my side. Christian has been working on development tools at SAP for quite a few years now. Like, for instance, the SAP HANA and ABAP development tools for Eclipse. And he is now a product owner for CAP tools and helps spread the word about CAP in general. Daniel has been with SAP since 2004. In the time since then, among other things, he invented and designed the Core Data Services, in short CDS in 2012, which got quite broadly adopted all across SAP in HANA, ABAP and now in CAP, as well as designed or co-designed - will maybe clarify that - the Cloud application Programming Model "CAP" in 2017, which he now leads as a chief architect and chief product owner. Besides, as he claims, still being a notorious developer. Now there is one thing I always struggle with when I speak English to Germans. Christian, would you prefer to be addressed as Christian or as Christian? I might mix it anyway.

Christian: I don't care, actually, I don't mind. So, whatever, like, is easier for you and you think your audience will like.

Karsten: Daniel, would you prefer to be addressed as Daniel or Daniel?

Daniel: Whatever you prefer as well.

Karsten: Okay, so then I can mix as I choose. So, Christian addressing you in the German pronunciation. Now, what is CAP? Can you give us the short definition?

Christian: Short definition. Usually, we do that with a little like demo itself, right? But here that we are now on the podcast, it's just words. So, it's basically to help people build applications in the enterprise context, right? It's all about application development and developers. CAP itself, technically, it's like an SDK that's probably like the most usual term: SDK, a framework, or sometimes we call it a programming model here. It's a set of libraries and tools both for Node.js and for Java. In its core CAP has a little language that we invented called Core Data Services, CDS in short, that helps us do various things. It, for instance, helps you expose your API definitions as a REST or OData service. It helps you phrase and model your data model. It translates queries to SQL, etc. And with just few lines of code, you can basically have a whole, a complete application running that serves OData request and that has its model deployed to a database. So that's just a little excerpt for from what of what CAP can do.

Karsten: All right, so I can tell there is at least a context to CDS, the Core Data Services, as far as CAP's concern, Daniel, is there anything else that we should put CAP into context with?

Daniel: Yeah. So CDS, as you mentioned, is what we call our ubiquitous modelling language. We use it to model data structures, data models, services and even UIs. And basically, this is also how to explain where CAP actually resides in this well known stack of an application, consists of a data layer, a logic layer or service layer, you could say, and presentation fronted layer. And in the SAP world, basically the UI layer to start with that, is usually addressed with Fiori Elements technology of UI5 technology, as we say. And this is also the case in CAP. So, CAP very much supports Fiori-based, UI5-based front ends even by CDS specific annotations. And on the database side, SAP has SAP HANA as the recommended powerhorse database. And this is also what we support from CAP. We deploy CDS models directly to HANA and then have an out-of-the-box support of the HANA database, for serving queries. And the rest is the logic layer, the service layer. And this is actually where CAP contributes our so called CAP runtime or SDKs, which fill in the missing gap between the front-end Fiori and the back-end HANA, being the service layer served by CAP. And maybe to add, if you know SAP, then you likely know ABAP and that's ABAP in the on premise world and times and even in the cloud as well. That same structure ABAP serves HANA, serves Fiori UIs very much, and it has the middle layer served by the ABAP server and the ABAP processes. And this is where CAP resides, and then in an ABAP world.

Karsten: Okay. So, knowing a bit about what CDS is and where all it's deployed and everything else you've mentioned, that sounds pretty wide reaching. Which applications already run on CAP? And can I like if I look at an application, can I tell it's based on CAP or Where would I have to look to be able to tell that?

Daniel: No. Yeah. Not by looking at it. It looks in the browser as any other application, especially when it comes to Fiori Elements based ones. But CAP is in the backend server part, of course. And what we do have is really strong adoption in SAP internal application projects. Actually, CAP is now also rolled out and promoted as the go to framework in the non ABAP world, RAP in the ABAP world. And we have about 100 plus projects in SAP internally, big projects. One of them, which is out in live already is advanced financial closing, for example. And yeah, the number grows steadily. So, adoption is really strong.

Karsten: Now, you've mentioned one perspective that is ABAP non-ABAP. Let me switch to another one. Is CAP a framework that is made for SAP made applications only. So only for SAP internal development? Or how is that.

Daniel: Oh, well, we won't be in this podcast about Open source, I think if it was for internal projects only. Yeah, actually from the very beginning we made available internally as well as externally, when we were on TechEd 2018, which was one year after we started CAP, we made it available to external customers immediately. And actually, the interest by external customers and partners also drove the internal adoption to very much extent. So, definitely not only for internal projects even I would say it was external first.

Karsten: And I confess, I knew and as you said it, you wouldn't be in this podcast if it was internal only. How would it be open source then even because this is really why we're here today. That now CAP is being gradually open sourced, right?

Daniel: Right, exactly.

Christian: So, we actually got requests quite early on from the community, as soon basically as we made it available externally. There was like some early enthusiasts saying - Hey, this would be cool, like to contribute or to evolve further, right? With the community, by the community. Why don't you make it open source - And it's like these discussions forth then back also internally, you know, the strategic discussions here internally is it good or should we do it, etc., etc. But here we are now, like having decided for that. And then the question was, okay, what's our approach here? In the meantime, CAP had grown quite significantly in terms of code base and usages and customers. So, it's not just easy to put it out like that, right? So, we had to basically come up with an approach. And that approach is not a big bang, but like start from the periphery approach, where we basically said, okay, let's start from packages that are easier to put out, easier to consume, easier to contribute to, cleaner in the API first, and then gradually from there find our way through the core parts of it, which will come later. So, we start basically from like a protocol adapter like GraphQL and database connectors like, PostgreSQL, because as I said, this is probably where the community anyway want to contribute and help us with. And then the core parts, like the compiler part, the core runtime, this will come later. And there's of course also always like proprietary parts like HANA connectors, HANA drivers that probably will never be published because it doesn't make too much sense. Yeah, so that's our approach here.

Karsten: So, if I understand correctly, the gradual opening starts or progresses over the two sides. One is the bottom, basically the database side and the other one is the consumption side protocols, UI frameworks and so on, right?

Christian: Exactly.

Karsten: Okay. And that means the community is now - let's take the database side first - is invited to do what?

Christian: Basically, to help us shape the APIs, the interplay between, like, the core parts and that adaptor. And then also to, like, really help us implement with all the expertise, for instance in Postgres, Postgres world and contribute, like, bug fixes and documentation maybe etc. So, this is where they will help us. And then they probably also, soon come up with new adapters, new connectors. They even did, someone in the past, like an adapter without us, right? Without a proper API. They already came up with working version and alpha version, so to say, of a Postgres adapter, which was quite cool. And we didn't anticipate that at the time.

Karsten: Just to get that correctly. Now, you mentioned Postgres. Is there already an existing connector or would that exactly be one of the things that would now in the future be community produced?

Christian: So, there is an existing one which I would call like a whatever alpha etc. version of beta maybe, and that is like now reshaped and, basically a new major version done again with the community now like a proper APIs and cleaner separation etc. So next evolution so to say.

Karsten: Okay. And then then maybe switching to Daniel again, and switching to the other side of things on protocols and more consumption side, I assume that something like OData stays closed, and others are the ones open, or how is that?

Daniel: So, GraphQL, it's already open. We made this open source at TechEd last year, and OData, no, I think we are also going to open source the OData protocol adapter because OData is an open protocol, a standardized protocol. Yes, there is some quite SAP-specific extensions to OData, like the Fiori annotations, and indeed we might have a derivate of the OData protocol adapter, which I would call the Fiori protocol adapter. And that one then likely would stay proprietary to SAP.

Karsten: And how about totally different things? I hope I'm getting the analogy right. Would it be open to also connect to, say, Google Angular?

Daniel: Yeah, of course: Google Angular is using a REST-based protocol, but of course it would also benefit from, say, data bindings that we could standardize and things like that could be provided also by the community. It's not on our radar screen for today at least. So, it's an invite to the community.

Karsten: Sounds cool. And that's basically also the two sides of things. The one is the database side and the other one is the protocol side. Or is there anything else in the ways of open sourcing that's worth mentioning, Christian?

Christian: Well, we should mention, our sample code repositories that we have there since I think 2019. So quite some time already providing code samples, smaller and bigger ones to the community, both for node.js and for Java. There's even like an application that focuses on rather on the Fiori side, like showing you how to basically build a Fiori Elements application on top, which is called "SFLIGHT". So, people might know that from the ABAP times, where the infamous famous SFLIGHT application was there and it's basically now reborn, for the cloud and the CAP world. So that is out there, and there's like other packages, sample code packages there. It's all listed on our homepage. And then the next thing we plan is documentation, right? A much demanded package that people want to basically contribute. Maybe just like typical a typo bug fix, but like also bigger things, new guides, etc. and we plan to do that in the next months. We are at the moment about to split internal documentation from the external one, in a more clear fashion and then we'll put the documentation out. And then we gradually, as mentioned before, gradually find our way through the periphery towards the CAP core, you know, core runtime, Node.js runtime and also the compiler that does all that magic translation between like the CDS model to SQL, etc. So, these parts would be also come in parts at least somewhere in the future. Right? So that's the list of things, the bill of material.

Karsten: And I guess that should at least as a first impression cover as well what CAP is and does and also what the plan for open sourcing is. Let's maybe then examine a very different question: Why are we doing this? I mean, how does it benefit the community - that's maybe the most obvious. How does it benefit the SAP CAP team? And how does it benefit SAP overall?

Daniel: Yeah, that's a pretty good question. And actually, this is the question we asked us in the very beginning and also before taking the decision of making it open source. So maybe first of all, it's not to just get this label of, "Hey, we're open source, so we are cool". There has to be a real benefit for both the community and for us as SAP and the CAP team. Now, maybe starting with the community, putting me in the shoes of a community, customer, or partner. I think one important thing is that they are able to have more influence on the direction of how we develop things by joining our initial setup, things, the actual development, maybe even have pull requests that add new features to what we do. And this is, of course, where open source really shines. The other thing is also when we maintain and support things, so think of currently a customer found a bug or ran into something they would have to open a message on answer.sap.com, a very chat-like thing. Much better would be if they could just open pull request in GitHub or open an issue in GitHub and refer to the sources that we have to work on. So, we expect both. This is now going to our choose the CAP team to choose. We expect the community to have more simpler ways to contribute and do bug reports. And for us at the CAP team to have less efforts in reproducing things in compared to answer.sap.com messages. And overall, I think having this software open source would attract developers. Developers nowadays are just used to have things open source. So, we hope to attract a larger share of the developer market, which is important for us as SAP, as well as partners and customers to find developers which have then CAP knowledge.

Karsten: Let me maybe jump one question back, though. Is the ultimate plan to have CAP fully open source? Or will we always on SAP side own this last castle in the core of CAP, basically?

Daniel: Yeah, that's related to licenses questions which are very sensible. Our ambition to have it, is to have it fully open source, everything at least which is not related to SAP proprietary technologies. So, also including the core of CAP. But that's a journey.

Karsten: Saying "our ambition", that's yours and Christian's and maybe then some colleagues. But you already disclaimed there's, of course, question of licenses and many other legal aspects maybe behind that. So, we will not make that promise here, right?

Daniel: Exactly. That's why.

Karsten: Cool, I think we can agree on that. And then in these first phases of opening it up, is that already working? Is there already activity in a community with what's available now, or is this all still a plan? I didn't quite get that.

Daniel: Yeah. So, when we look at the CAP samples which are out since over a year, as Christian said before, we see great contribution there, definitely. So, people really add new samples, fix our samples basically, add new flavours to the samples, like I remember the TypeScript contribution, and this is quite great. And yeah, it really turned out to be great for the CAP samples already. I think similar would be if we would have our documentation out, this would just be great as well. And for the database, the new database layers, most prominently the Postgres implementation. We currently do that in close collaboration with community members, prominent ones, and our teams. So, this is actually happening right now. Yes.

Karsten: Okay, but you're somehow phrasing that differently. Is the database side really open source yet or not quite?

Daniel: Yeah, the new database layer of Node.js and in particular the Postgres implementation that is currently already open source, yet still in the private repo because we are just onboarding all the people on that internally and externally. But we will just switch on.

Karsten: Private repo means by invitation only or...?

Daniel: Exactly, right.

Karsten: But will go probably eventually fully open or...?

Daniel: It's actually very soon in the next two or three months.

Karsten: All right. And if I am interested now, as one of our listeners here, where is the best place to get started with learning even more about CAP or even contributing?

Christian: Yeah, I think that the best entry place is really our home page on cap.cloud.sap where all the information come together, both documentation and announcements about like events and release notes, which we put out every month. Usually, telling you about not only what features we've added, what bugs we fixed, but also which things have changed in the open source community like new packages available, etc. So that's cap.cloud.sap. I think that's the URL is so short that you can probably remember it. And then we have like the SAP community pages, where we have like a dedicated page there. And there's also a reCAP conference coming up on July 6th and 7th together with UI5con. So that will be quite exciting. And there we will probably share more details on like the status and the approach and the learnings in that time. So, reCAP 6th and 7th July 2023.

Karsten: Is reCAP, is that a real conference or still an online-only event?

Christian: It's an on-site conference indeed. So, this year, first time on site, last year it was like hybrid or completely virtual. And now we take the adventure and go live and on site, quite exciting.

Karsten: Where is it going to be?

Christian: Where? It's here in the Walldorf area. It's in basically in the neighbouring town of St. Leon-Rot?

Karsten: By the way, the links we will provide in writing on the podcast website as well. So cap.cloud.sap is yes, easy to remember, but no need, it will be on the podcast website. With that covered - I didn't announce it - but that was the traditional before-last question, the places to go. And the traditional last question is: If you would have three or four things to remember from this podcast - key takeaways one calls it - what would they be? If you want, go two each. I don't know, who will start?

Christian: I can maybe start. So, CAP is here to stay. It already has quite a central role in SAP. It receives a broad adoption. The whole, like, CDS thing is already quite known, definitely in the ABAP world, but now also in the cloud world. So, CAP is here to stay. That's one thing. And use CAP to develop business applications in the SAP ecosystem, but also outside it's the preferred framework and SDK here at SAP. So should be yours as well.

Karsten: Okay, thanks. Daniel, you have another key takeaway to add?

Daniel: Yeah: to Open source. Yeah, open-sourcing CAP means not to be cool with the label, as I said, but really you can engage with us and make us better and get better. So, it's really an invite to you to contribute, to work together with us on the future of CAP.

Karsten: All right.

Christian: Meet us at GitHub.

Karsten: Meet you at GitHub and meet you in July.

Christian: Yeah, that's true. That's true. We all there.

Karsten: Right. Daniel, you only had that one, which is, of course, as the general call to action, always a great one.

Daniel: Yeah, that's fine. Three to remember is the best.

Karsten: All right. Okay. Three to remember is the best. Let's finish with that tagline. So, thank you very much, Daniel. And thank you very much, Christian, for being our guest today. It was great to have you both here.

Christian: Thanks for having us.

Karsten: You are very welcome. And thank you out there for listening to the Open Source Way. If you enjoyed this episode, please share it, and don't miss our next episodes that are usually being published every last Wednesday of the month. You will find us on openSAP and also in most other places where you usually find your podcasts like Apple Podcasts, Spotify and so on. Thanks again, everyone. Thanks again, Christian and Daniel and bye bye.