0:0:0.0 --> 0:0:12.250  
Speaker 1  
I I can start by, uhm, basically introducing a little bit introducing to you a little bit the concept of or also the contest of our work.

0:0:12.340 --> 0:0:31.860  
Speaker 1  
So as I said, we we are targeting the conformance architecture conformance no and what we realized is that, yeah, in general you might have a situation let's say where you have maybe I can use this, yeah.

0:0:34.960 --> 0:0:41.120  
Speaker 1  
This image so it's umm I will share the screen.

0:0:42.870 --> 0:0:50.620  
Speaker 1  
So if you look here, uh, hope you can see the screen and then in a while you should see manage.

0:0:51.190 --> 0:0:52.330  
Speaker 1  
Yeah, this one.

0:0:53.490 --> 0:1:0.660  
Speaker 1  
So in general, what we realize is that during the development the software development, but also the architectural process.

0:1:0.670 --> 0:1:3.680  
Speaker 1  
Now you may have at 1.1 architecture.

0:1:3.690 --> 0:1:9.500  
Speaker 1  
OK, these architecture clearly can have some evolutions over time.

0:1:9.510 --> 0:1:20.720  
Speaker 1  
So you you start defining this architecture and then let's say refining the the architecture based on your increased knowledge or or new design decision requirement, whatever that may be.

0:1:21.780 --> 0:1:39.430  
Speaker 1  
Also what we know is that you never or rarely define an architecture from scratch, and you you often have some guidelines or even more abstract architecture that can in a way constrain the development of this architecture.

0:1:39.440 --> 0:1:40.530  
Speaker 1  
An example of this.

0:1:40.540 --> 0:1:49.210  
Speaker 1  
It could be, for instance, when you use reference architecture that give you like the context, the boundary and then you define more.

0:1:49.220 --> 0:1:53.540  
Speaker 1  
Let's say specifics software architecture for your case, but another another.

0:1:53.810 --> 0:1:56.840  
Speaker 1  
Let's say example could be software product lines.

0:1:57.100 --> 0:2:10.600  
Speaker 1  
When you have, yeah, the the software product line architecture and then you have the definition of your of your variance or even in general when you won't do describe an architecture that is compliant to a given style.

0:2:10.610 --> 0:2:46.590  
Speaker 1  
So let's say you want to use publish subscriber and or whatever other style and then this implies that you need to basically follow some rules and So what we know is that although this is are common practices currently in the architectural world, there are no processes that allow an architect an architect to, I mean derive this this more concrete architecture from this style software product line or whatever by having a process that continuously check the conformance.

0:2:46.600 --> 0:2:59.740  
Speaker 1  
No, and on the one hand, we realized that in, in a way, maybe conformance at at every steps is not needed, because at the beginning you don't want to be under person conformant, for instance.

0:2:59.750 --> 0:3:6.10  
Speaker 1  
But we also realize that it's important to have an indication on how far you can.

0:3:6.70 --> 0:3:12.250  
Speaker 1  
Let's say you can be from being conformant in specific domain.

0:3:12.650 --> 0:3:24.490  
Speaker 1  
This may be even more important, such as safety critical when you have to show some sort of conformance with the with the some uh safety regulation and this and that.

0:3:24.620 --> 0:3:48.650  
Speaker 1  
So in in short, what we did in order to let's say tag all this, we introduced the concept of continuous conformance as yeah a concept but also as a process that allows you to allows an architect to continuously basically have a measure of how far an architecture can be.

0:3:48.700 --> 0:3:48.990  
Speaker 1  
Uh.

0:3:49.0 --> 0:4:0.30  
Speaker 1  
With respect to a more abstract architecture that in a way dictates the the put some boundaries on on the definition of this more concrete architecture.

0:4:0.960 --> 0:4:3.140  
Speaker 1  
So is it clear until now my explanation?

0:4:5.540 --> 0:4:5.910  
Speaker 2  
So.

0:4:5.920 --> 0:4:7.0  
Speaker 2  
So let's put it like that.

0:4:7.10 --> 0:4:8.720  
Speaker 2  
I have few questions, let's clear them up.

0:4:9.410 --> 0:4:9.540  
Speaker 1  
Yes.

0:4:9.880 --> 0:4:16.710  
Speaker 2  
Are you focusing on some specific type of software like embedded or it's just general software architectures?

0:4:16.880 --> 0:4:18.770  
Speaker 1  
No, this is in, in in general.

0:4:18.780 --> 0:4:22.860  
Speaker 1  
So we are not focusing on any specific domain for the moment.

0:4:22.860 --> 0:4:23.590  
Speaker 2  
OK.

0:4:23.640 --> 0:4:26.590  
Speaker 2  
And then what is your definition for conformance?

0:4:26.600 --> 0:4:27.710  
Speaker 2  
Do what?

0:4:27.720 --> 0:4:31.690  
Speaker 2  
Not the the like a formal definition, but what do you mean when you say conformance?

0:4:32.130 --> 0:4:32.360  
Speaker 1  
Yep.

0:4:31.700 --> 0:4:34.290  
Speaker 2  
So you have something to conform to, but what's that?

0:4:38.60 --> 0:4:39.210  
Speaker 1  
Yeah, exactly.

0:4:34.300 --> 0:4:39.520  
Speaker 2  
Something the reference abstract architecture or what OK.

0:4:39.220 --> 0:4:50.690  
Speaker 1  
So it's the it's uh, in in the general case, if you have a uh, let's say 2 architecture, one is more abstract than one is more concrete, let's say then you will have the abstract architecture.

0:4:50.700 --> 0:4:55.290  
Speaker 1  
That is the architecture that you want to become formant to.

0:4:55.700 --> 0:5:14.220  
Speaker 1  
And as I said, it can be a reference architecture and you want to be conform and do this reference architecture or it can be architectural style and to be precise we have of course in the in the paper the definition of conformance mathematically.

0:5:14.230 --> 0:5:17.630  
Speaker 1  
But I mean, this is not the did ask.

0:5:16.410 --> 0:5:25.110  
Speaker 2  
Yeah, because that's what I understood that you want to quantify so that you be able to measure it so that you be able to show progress or regress towards something.

0:5:24.480 --> 0:5:25.410  
Speaker 1  
Yes, exactly.

0:5:46.580 --> 0:5:46.790  
Speaker 2  
2.

0:5:25.420 --> 0:5:51.720  
Speaker 1  
So we we defined the the conformance as a distance function basically so that quantifies exactly how far are you from the the A given architecture and for the moment of course we look at the, I mean maybe we can start with the example that I think clarifies a little bit also what we what to meet what we mean because we can actually discuss on something more practical.

0:5:51.930 --> 0:6:18.770  
Speaker 1  
So if you look at this this slide and this is the the tool that we, I mean that embodies all this process that we discuss and to start with OK on the left side you have let's say a navigation panel from which you can load any sort of as we say the reference architecture style or whatever you had in the knowledge base.

0:6:18.850 --> 0:6:22.370  
Speaker 1  
So let's say that you choose a reference architecture.

0:6:22.380 --> 0:6:33.430  
Speaker 1  
In this case, for the sake of discussion, is very simplified and this is the graphical representation of the reference architecture and also in this reference architecture you have 3 components and.

0:6:33.440 --> 0:6:35.730  
Speaker 1  
These three components communicate in this way.

0:6:35.740 --> 0:6:51.460  
Speaker 1  
So essentially, for instance, what we can say is that this reference architecture already put some constrained because according to it you cannot have connection between B&C for instance because would be illegal, no.

0:6:52.470 --> 0:6:55.10  
Speaker 1  
Again, this is very simplified case so.

0:6:54.900 --> 0:6:56.180  
Speaker 2  
Can you make it the full screen?

0:6:56.190 --> 0:6:58.910  
Speaker 2  
The video please, like I really have trouble seeing it.

0:6:58.630 --> 0:6:59.580  
Speaker 1  
Yes, yes, yes.

0:6:59.350 --> 0:7:0.120  
Speaker 2  
It's not possible.

0:6:59.660 --> 0:7:3.890  
Speaker 1  
Uh, it's just that if I go on uh on full screen then.

0:7:6.290 --> 0:7:6.610  
Speaker 1  
Let me.

0:7:9.560 --> 0:7:11.740  
Speaker 1  
Yes, I can also zoom a little bit.

0:7:12.830 --> 0:7:14.470  
Speaker 2  
OK, that's much better.

0:7:14.780 --> 0:7:15.990  
Speaker 1  
Yeah, that's I.

0:7:16.0 --> 0:7:19.150  
Speaker 1  
I think I will mess up a little bit with the common, but it's fine.

0:7:19.320 --> 0:8:0.360  
Speaker 1  
So and and anyhow, this is the graphical representation of it which is loaded the and and it's given to you for yeah simplicity and then what you do is that you start to create your architecture and to do so we and provide this, let's say textual editor, where is essentially what you do is that you write down the components and uh you the the component of your let's say architecture concrete architecture and you also specify to which component of the reference architecture they map.

0:8:0.470 --> 0:8:9.260  
Speaker 1  
So as I don't know if you can read, but here you have like a name 81 and implements a capital A1.

0:8:9.540 --> 0:8:16.330  
Speaker 1  
While you do this, uh, the tool gives you on the one hand the graphical representation of your architecture.

0:8:16.340 --> 0:8:19.710  
Speaker 1  
Not that you can see here in the in the bottom bar.

0:8:19.720 --> 0:8:30.550  
Speaker 1  
Now it's cut and also it starts to give you the basically measure of the conformance in these two.

0:8:32.950 --> 0:8:39.560  
Speaker 1  
In these two tabs over a year, so in the console, uh we show event possible violations.

0:8:39.570 --> 0:8:49.850  
Speaker 1  
So if you have a violation in your architecture, are written here to actually and in this conformance view tab you will see stand of.

0:8:49.950 --> 0:8:53.620  
Speaker 1  
I mean a user will see stand a radar chart.

0:8:53.730 --> 0:9:1.710  
Speaker 1  
So now I can, if we go on with the video, we can show, for instance, one example of violation.

0:9:2.630 --> 0:9:3.840  
Speaker 1  
So now here we are.

0:9:4.800 --> 0:9:8.290  
Speaker 1  
Uh, adding a new component?

0:9:8.300 --> 0:9:11.610  
Speaker 1  
A component B1 that you see you will see created here.

0:9:11.880 --> 0:9:15.470  
Speaker 1  
Now the component be one is still not mapped to any component of.

0:9:15.620 --> 0:9:20.670  
Speaker 1  
Let's say the the reference architecture and this is in a way allowed.

0:9:20.680 --> 0:9:29.30  
Speaker 1  
And then we are also adding a connector and we can go on a little bit with the video.

0:9:31.110 --> 0:10:22.920  
Speaker 1  
O basically at one point, yes year we are that the uh in short we add that a component B in our be one in our architecture our component C1 and then we added a connector between B1 and C1 and as soon as we do that we get and we save we get this violation asset to all of you in the console because this kind of connector is not legal according to the reference architecture OK this very simple case and the the same time you get this this view this rather view that show it's show it graphically so this is the basic basic use and I don't know maybe you're ready you still have question on on on this.

0:10:22.990 --> 0:10:23.130  
Speaker 1  
Yes.

0:10:27.470 --> 0:10:29.0  
Speaker 2  
No, I don't have a question.

0:10:29.10 --> 0:10:31.450  
Speaker 2  
I I can see your point here and what you're doing.

0:10:34.290 --> 0:10:40.0  
Speaker 2  
I'm trying to wrap my head around it and connected to something more practical.

0:10:40.10 --> 0:10:40.730  
Speaker 2  
Let's put it like that.

0:10:41.620 --> 0:10:41.820  
Speaker 1  
Yeah.

0:10:41.720 --> 0:10:49.170  
Speaker 2  
Uh, so I'm wondering why you strictly say that if something so in this case it's something added.

0:10:49.300 --> 0:10:49.870  
Speaker 2  
OK.

0:10:49.880 --> 0:10:50.850  
Speaker 2  
How do we define it?

0:10:50.860 --> 0:10:59.300  
Speaker 2  
So you said that in your reference architecture you have 3 components AB and C and you say the communication between B&C is not allowed directly.

0:11:3.980 --> 0:11:4.180  
Speaker 1  
Yeah.

0:10:59.310 --> 0:11:8.570  
Speaker 2  
Everything has to go through a, let's say a can be some kind of like a man in the middle that has some validation or whatever properties.

0:11:9.320 --> 0:11:13.320  
Speaker 2  
And then if you bypass that, then you say it's a violation.

0:11:13.400 --> 0:11:18.400  
Speaker 2  
So you basically say everything that is not specified in the architecture is a violation.

0:11:19.630 --> 0:11:19.850  
Speaker 1  
Yeah.

0:11:20.550 --> 0:11:22.50  
Speaker 2  
That's the definition, OK.

0:11:21.220 --> 0:11:24.490  
Speaker 3  
Yeah, we did respect to the reference architecture course.

0:11:24.60 --> 0:11:27.90  
Speaker 2  
Yes, everything that is not defined in the reference architecture.

0:11:27.200 --> 0:11:27.430  
Speaker 3  
Yep.

0:11:27.180 --> 0:11:31.600  
Speaker 2  
So any extra element is a violation of the reference.

0:11:32.420 --> 0:11:33.500  
Speaker 3  
Umm, no, you.

0:11:31.930 --> 0:11:33.760  
Speaker 1  
Yeah, or any means used.

0:11:33.880 --> 0:11:34.690  
Speaker 1  
Misused.

0:11:34.700 --> 0:11:39.910  
Speaker 1  
Uh, let's say yeah, element of the reference architecture can be a violation as well.

0:11:40.640 --> 0:11:40.860  
Speaker 2  
OK.

0:11:40.960 --> 0:11:51.180  
Speaker 3  
But you can you can have the components that are not considered the reference after that or not because for example suppose that you want to implement the component.

0:11:51.510 --> 0:12:6.520  
Speaker 3  
The uh see enough capital in the reference you can implement by using the two component concrete the architecture so it it is allowed no in general right.

0:12:5.970 --> 0:12:6.660  
Speaker 2  
Umm OK.

0:12:7.700 --> 0:12:14.320  
Speaker 3  
Great, but you are too big a conformance with the reference architecture.

0:12:17.510 --> 0:12:17.760  
Speaker 2  
Yeah.

0:12:14.330 --> 0:12:27.190  
Speaker 3  
No, this case you have to to connect the components that implements a capitalized with respect to be capitalized, for example.

0:12:29.930 --> 0:12:32.280  
Speaker 2  
Yeah, it's basically anti pattern what we saw.

0:12:37.880 --> 0:12:38.980  
Speaker 3  
Yeah, yeah, yeah, yeah.

0:12:32.290 --> 0:12:42.380  
Speaker 2  
Now that you have, if you think about the reference architecture as a pattern, then what you show example is an anti pattern that you detected OK.

0:12:42.220 --> 0:12:42.690  
Speaker 3  
Yeah.

0:12:41.870 --> 0:12:43.640  
Speaker 1  
Yeah, in this in this symbol.

0:12:43.730 --> 0:12:44.10  
Speaker 1  
Yeah.

0:12:44.20 --> 0:12:46.560  
Speaker 1  
Except yeah, but.

0:12:45.60 --> 0:12:46.720  
Speaker 2  
Yeah. Good.

0:12:42.740 --> 0:12:46.840  
Speaker 3  
Yeah, it is a guidelines. Umm.

0:12:55.880 --> 0:12:56.510  
Speaker 2  
I lost you.

0:12:56.520 --> 0:12:57.130  
Speaker 2  
I cannot hear you.

0:12:58.10 --> 0:13:0.40  
Speaker 3  
OK, now, yes.

0:13:0.980 --> 0:13:1.220  
Speaker 2  
Yep.

0:13:0.150 --> 0:13:2.90  
Speaker 3  
Uh, yeah, yeah, no, yes.

0:12:59.890 --> 0:13:2.270  
Speaker 1  
Can you hear me OK?

0:13:2.280 --> 0:13:20.930  
Speaker 1  
I touched the the MIC by so I was saying that this is this is the base case of course, but also as you can see here in our maybe you can, but I'm telling you in this right side of the screen there is another window where that we call it, let's say script validation script.

0:13:20.940 --> 0:13:35.540  
Speaker 1  
But essentially what we allow is also the definition of extra constraint that you can boot on the reference architecture and those constrained cannot will basically enrich the conformance check.

0:13:35.630 --> 0:13:36.750  
Speaker 1  
So here you can boot.

0:13:36.760 --> 0:13:37.180  
Speaker 1  
I don't know.

0:13:37.190 --> 0:13:53.700  
Speaker 1  
Whatever kind of logical constrained you want on, for instance, the you can say, OK, be capital needs to be instantiated exactly from one component or only one or this communication must be in a given way something like this.

0:13:53.710 --> 0:14:4.780  
Speaker 1  
So whatever you need to reach in a way, the also the semantic of your reference architecture and this will be checked from the conformance. Uh.

0:14:4.960 --> 0:14:5.220  
Speaker 2  
Good.

0:14:6.700 --> 0:14:6.850  
Speaker 1  
Yeah.

0:14:6.390 --> 0:14:8.980  
Speaker 2  
And this is not part of the reference architecture.

0:14:12.800 --> 0:14:13.610  
Speaker 3  
Yeah. Yeah, yeah.

0:14:13.680 --> 0:14:14.900  
Speaker 3  
Yeah, because.

0:14:8.990 --> 0:14:15.660  
Speaker 2  
This is used in combination with the reference architecture to who, hence the put further enhance the constraints.

0:14:12.280 --> 0:14:24.590  
Speaker 1  
Yes, exactly because you you may have this is, uh, we we did this for, let's say, different part poses one is because in general you.

0:14:26.530 --> 0:14:39.660  
Speaker 1  
In the general case, you may have something, uh, like, uh, a very abstract like it can be a style, or it can be again a reference architecture for most cases or different domains.

0:14:39.670 --> 0:14:49.960  
Speaker 1  
But then you may still want to like customize it a little bit on your on your product or your whatever project or whatever.

0:14:50.150 --> 0:14:57.670  
Speaker 1  
And that's why we we we splitted in in in these two let's say stage is let's call it like this.

0:15:3.880 --> 0:15:4.260  
Speaker 1  
Yeah, you.

0:14:58.520 --> 0:15:13.230  
Speaker 3  
Because typically an architecture is not only components and the connect or not, because you have some architectural decisions and this is a view where you can specify this architectural decision.

0:15:13.580 --> 0:15:13.870  
Speaker 3  
OK.

0:15:14.630 --> 0:15:15.270  
Speaker 1  
Yeah, exactly.

0:15:15.280 --> 0:15:15.990  
Speaker 1  
There's also another.

0:15:15.580 --> 0:15:17.320  
Speaker 3  
You think they may not be up?

0:15:21.730 --> 0:15:41.160  
Speaker 1  
But now if this is clear in our way, what we also will, I mean I think the the the interesting part is the fact that we OK this is a the best case now when you want to do this conformance check but then to put this in relation with the figure that we show before.

0:15:41.290 --> 0:15:52.600  
Speaker 1  
So meaning when you may have a error key of of, let's say architecture, each of which may be conformance with the more abstract one.

0:15:52.750 --> 0:16:4.200  
Speaker 1  
Then, in short, what we presented over a year, uh can be extended, can be used at any level of abstraction, because simply what?

0:16:4.450 --> 0:16:6.740  
Speaker 1  
Let's let's give you an example.

0:16:6.970 --> 0:16:26.370  
Speaker 1  
If this that we are developing here this reference architecture, it can be a style then uh we we develop our reference architecture over here and once we develop our reference architecture we can save this architecture and we can reuse it in the in the tool.

0:16:26.660 --> 0:16:34.310  
Speaker 1  
So that in the next, let's say stage of the OR next iteration is stand of having this.

0:16:34.380 --> 0:16:36.910  
Speaker 1  
Let's say let's call it style.

0:16:36.920 --> 0:16:44.730  
Speaker 1  
We will have the newly defined reference architecture based on which we can define a more concrete architecture runs on and so forth.

0:16:44.740 --> 0:16:53.780  
Speaker 1  
So we can go at 10:11 of the abstraction and we without changing basically anything the notation and and all the things would be the same.

0:16:54.700 --> 0:17:6.50  
Speaker 1  
And this is 1 Bart, but also another part is that we can, uh, basically uh define uh different.

0:17:6.60 --> 0:17:7.140  
Speaker 1  
So when we are.

0:17:9.280 --> 0:17:15.450  
Speaker 1  
Checking the conformance, we can also be more or less, uh.

0:17:15.460 --> 0:17:16.290  
Speaker 1  
Restrictive.

0:17:16.340 --> 0:17:24.50  
Speaker 1  
So we maybe want to check the conformance only with respect a given part of the starting reference architecture.

0:17:24.140 --> 0:17:27.280  
Speaker 1  
In that case, we allow that and we allow that.

0:17:28.600 --> 0:17:44.840  
Speaker 1  
In a simple way, that is, you basically take a the, the, the component and connector that you are defining your architecture and and you simply do not map them to uh element in the reference architecture.

0:17:45.270 --> 0:17:55.890  
Speaker 1  
When you do that, the, uh, basically the uh, the engine will not the conformance engine will not consider this element that are not mapped or implement.

0:17:55.930 --> 0:17:56.930  
Speaker 1  
Yeah, they are not mapped.

0:17:56.940 --> 0:18:3.20  
Speaker 1  
Let's call it like this, and then the conformance with only be partial in the in the.

0:18:3.90 --> 0:18:6.940  
Speaker 1  
Yeah, in the part of the reference architecture that is actually instantiated.

0:18:8.230 --> 0:18:11.950  
Speaker 1  
Also, I don't know if I was, uh, clear enough with the this explanation.

0:18:15.860 --> 0:18:16.40  
Speaker 1  
Yeah.

0:18:13.160 --> 0:18:16.720  
Speaker 2  
I I can tell you how I understood it and maybe you can judge on that.

0:18:17.220 --> 0:18:26.30  
Speaker 2  
What I understood is, let's say you're reference architecture is too high level and then you have some refinements done down the process which are not captured by the reference.

0:18:26.520 --> 0:18:33.240  
Speaker 2  
Then you will let me know as a user that it's a partial conformity because there are parts that you cannot assess.

0:18:34.510 --> 0:18:35.810  
Speaker 1  
Yeah as well.

0:18:35.820 --> 0:18:37.100  
Speaker 1  
Uh, yeah, exactly.

0:18:37.210 --> 0:18:47.840  
Speaker 1  
And another thing is that if you, uh, deliberately let's say have a very complex reference architecture or very not complex, but rather big reference architecture.

0:18:48.790 --> 0:18:51.360  
Speaker 1  
Uh, and uh?

0:18:51.430 --> 0:19:5.540  
Speaker 1  
You want to not check the conformance of your architecture with respect to the whole reference architecture, because you know that for instance you are developing just a part of subsystem or whatever.

0:19:5.630 --> 0:19:6.800  
Speaker 1  
So you can do that.

0:19:6.810 --> 0:19:20.600  
Speaker 1  
So imagine now again, this is a trivial example, so but imagine that you have ABC, OK and you are developing a subsystem which only will only target combo components, capital A and capital B.

0:19:20.770 --> 0:19:57.430  
Speaker 1  
So what you can do is that while you design your architecture, you will design your your your subsystem and simply not instantiated capital C by not doing that by I mean not distancia DE Capital C The conformance engine with only checked the conformance of what you're modeling with respect this part of the this sub model of or sub architecture consisting of capital A and capital B and this is a because of this yeah iterative and also agile process that you don't want to have maybe everything up front.

0:19:57.490 --> 0:19:59.300  
Speaker 1  
So this was the.

0:19:58.630 --> 0:19:59.700  
Speaker 2  
But the will I have?

0:20:2.260 --> 0:20:2.440  
Speaker 1  
Yeah.

0:19:59.710 --> 0:20:3.200  
Speaker 2  
Uh, let's say I'm using only NB and I have it correct.

0:20:3.210 --> 0:20:5.90  
Speaker 2  
What will be my conformance 100%?

0:20:7.870 --> 0:20:8.50  
Speaker 3  
Yep.

0:20:5.820 --> 0:20:9.580  
Speaker 1  
On A and B, yes, of course not on.

0:20:9.250 --> 0:20:13.940  
Speaker 2  
What will be my conformance if I have a component D and only D in my architecture?

0:20:14.70 --> 0:20:19.720  
Speaker 2  
If let's say my implementation has nothing to do with the reference architecture and I try to do compliance.

0:20:21.10 --> 0:20:21.740  
Speaker 1  
Uh no.

0:20:29.250 --> 0:20:29.470  
Speaker 2  
Yeah.

0:20:30.70 --> 0:20:35.360  
Speaker 1  
Then they zero because you don't have anything instantiated, so you won't.

0:20:35.490 --> 0:20:36.410  
Speaker 1  
Won't be checked.

0:20:36.590 --> 0:20:38.240  
Speaker 1  
Yeah, you don't have anything to check.

0:20:38.360 --> 0:20:41.110  
Speaker 2  
OK, so the conformity is 0 in that case.

0:20:41.320 --> 0:20:41.520  
Speaker 1  
Yeah.

0:20:42.260 --> 0:20:42.880  
Speaker 2  
OK, good.

0:20:44.390 --> 0:20:50.540  
Speaker 1  
And also we we allow we let's say another two another.

0:20:52.740 --> 0:21:14.480  
Speaker 1  
Uh, how can I see like feature that we are of the process is that we realize that again if we rollback at this image over here you at different stages of the development, uh, you may not want to be a using the same level of strictness for the conformance check.

0:21:14.530 --> 0:21:34.940  
Speaker 1  
OK, so very trivially at the beginning when you are in the, let's say, early face is maybe you want to brainstorm more, you don't want to be like very strict, but when you are going towards code or production, you want to have something that is I mean a conformance check that is more strict.

0:21:35.410 --> 0:22:13.790  
Speaker 1  
So what we allow or what we another feature is that we allowed to, uh, have a conformance check that are more or less less let's say St and we do this by of course this was another reason why we introduced this let's say possibility of specifying constraints by playing with this constraint that you can add and this constraint can of course uh again decide the strictness of your conformance, your conformance check at given at a given step of the development.

0:22:16.650 --> 0:22:23.340  
Speaker 1  
And of course, yeah, we have, like, a couple of example of uh, yeah.

0:22:23.590 --> 0:22:34.900  
Speaker 1  
Partial conformance checking, yes, and also the other things that I told you, but this is just graphically showing what I just told you.

0:22:35.350 --> 0:22:45.670  
Speaker 1  
Meaning that you are you define a component that will not be instantiating any component then yeah, you will not have issues basically.

0:22:47.80 --> 0:22:51.0  
Speaker 1  
So these were the main, let's say, feature of the.

0:22:53.30 --> 0:23:1.660  
Speaker 1  
Like the process that we try to embody it in this, uh, tool that in our opinion can help uh achieve in this continuous conformance.

0:23:2.90 --> 0:23:23.640  
Speaker 1  
And now, of course, we would like to, I mean, we already discussed started the discussion, but we will also have I mean we like to have your your view based on your experience on for instance whether this may be relevant or not the uh like possible uh limitation that you saw possible.

0:23:25.350 --> 0:23:26.460  
Speaker 1  
Yeah.

0:23:26.930 --> 0:23:27.940  
Speaker 1  
Yeah. Possible.

0:23:27.950 --> 0:23:29.950  
Speaker 1  
Uh, uh.

0:23:29.990 --> 0:23:37.670  
Speaker 1  
Good uses that you, uh, see of or you meeting up of this in your uh, daily work.

0:23:39.520 --> 0:23:41.810  
Speaker 1  
If any, of course, it doesn't have to be.

0:23:42.320 --> 0:23:43.780  
Speaker 1  
Uh, yeah.

0:23:48.200 --> 0:23:54.70  
Speaker 2  
I mean UMII can see several ways in which this can be meaningful.

0:23:54.600 --> 0:23:54.760  
Speaker 1  
Yeah.

0:23:54.760 --> 0:24:8.40  
Speaker 2  
1st and most obvious one, if you need to make an argumentation that you adhere to specific architectural pattern, let's say that you're building some safety critical system and you have to follow whatever pattern you have.

0:24:9.110 --> 0:24:10.60  
Speaker 2  
Then you can.

0:24:10.290 --> 0:24:15.240  
Speaker 2  
This can be good way to show that yes, I'm doing this and then I'm conforming.

0:24:15.250 --> 0:24:26.220  
Speaker 2  
I'm not conforming and for the things that you're not conforming, maybe you can give some argumentation why you're not conforming and so on so forth helps that you have a quantified value to say that I have.

0:24:26.230 --> 0:24:30.600  
Speaker 2  
I'm like X amount of conformant. Umm.

0:24:32.970 --> 0:24:35.50  
Speaker 2  
2nd is maybe like uh.

0:24:37.500 --> 0:24:42.470  
Speaker 2  
In a very complex scenario, when you do some change then you can get like an early feedback.

0:24:42.480 --> 0:24:48.790  
Speaker 2  
If you are, let's say if if you have a big architectural pattern but then small part of it is concerning.

0:24:50.920 --> 0:24:51.60  
Speaker 1  
Yeah.

0:25:0.680 --> 0:25:0.900  
Speaker 1  
Yeah.

0:24:48.800 --> 0:25:4.480  
Speaker 2  
Let's say the security when you have communication between components like I said here for instance, I magine component A to be something that makes sure that B&C talks in a in some specific follow some specific protocol, let's call it like that.

0:25:5.290 --> 0:25:8.80  
Speaker 2  
Then you can find those kind of subtle things.

0:25:8.90 --> 0:25:20.160  
Speaker 2  
Maybe you can find potential security flaws in in your architecture by doing this automated checks in a sense that you won't always component date to talk through component B or the other way around doesn't.

0:25:20.700 --> 0:25:20.900  
Speaker 1  
Yeah.

0:25:22.610 --> 0:25:26.300  
Speaker 2  
That can be an indicator, but that's a quite high level.

0:25:27.840 --> 0:25:34.120  
Speaker 2  
So basically when you and I, I would say that it's kind of on the architectural level to analyze certain things.

0:25:35.980 --> 0:25:36.800  
Speaker 2  
Uh. When?

0:25:38.500 --> 0:25:52.470  
Speaker 2  
When you want to design when you want to analyze the ARCHITECTURE of your system, but not the implementation like, you cannot really make a I cannot see how I can make a completely transferable to the implementation part.

0:25:52.480 --> 0:25:56.470  
Speaker 2  
It's only like how I structured how I envision, how I architect my system.

0:25:56.860 --> 0:26:16.430  
Speaker 2  
So in that respect, the security argumentation for following certain pattern, and of course the impact analysis like OK, what happens if I remove this component like how it will break my architecture, how compliant will I be so that goes along the lines of the de factor.

0:26:16.930 --> 0:26:17.110  
Speaker 1  
Yeah.

0:26:17.210 --> 0:26:18.520  
Speaker 2  
This could be quite useful.

0:26:18.530 --> 0:26:20.850  
Speaker 2  
Imagine this as a set of unit test cases.

0:26:20.860 --> 0:26:27.460  
Speaker 2  
When you have your code put in place, you know everything is working and then now you use this reference architecture.

0:26:40.410 --> 0:26:41.670  
Speaker 1  
Like you know something?

0:26:27.570 --> 0:26:43.570  
Speaker 2  
If it's detailed enough at some very low level of abstraction, then you can say OK, I remove this component, I do the analysis and I still analysis shows that I'm, let's say you have a threshold of at least 90% compliant and you say OK, I remove certain stuff and I'm still compliant.

0:26:43.900 --> 0:26:44.30  
Speaker 1  
Yeah.

0:26:43.580 --> 0:26:46.270  
Speaker 2  
So I can see potential value in that.

0:26:47.840 --> 0:26:53.710  
Speaker 1  
Uh, I mean our question, do you, I mean like in your work have?

0:26:54.350 --> 0:26:55.880  
Speaker 1  
And uh.

0:26:56.380 --> 0:27:10.520  
Speaker 1  
Had the need of the doing anything similar to what we just discussed at all or like or how often that's with this way you may need to do such such things.

0:27:13.310 --> 0:27:14.780  
Speaker 2  
That's a that's a good question.

0:27:17.120 --> 0:27:19.110  
Speaker 2  
In my narrow context of work.

0:27:21.420 --> 0:27:32.270  
Speaker 2  
I do these things quite often, but the the product that I'm mostly working with is not that complex that I need specialized tool support.

0:27:33.440 --> 0:27:34.50  
Speaker 2  
Umm.

0:27:34.240 --> 0:27:44.590  
Speaker 2  
And still in a sense architecture for me it's just like a guiding star, but not some artifact that I need to put a lot of emphasis on.

0:27:45.200 --> 0:27:46.830  
Speaker 2  
So in a sense, it's the code.

0:27:46.840 --> 0:27:47.870  
Speaker 2  
It's the testing.

0:27:47.880 --> 0:27:50.490  
Speaker 2  
It's the final execution that matters.

0:27:50.710 --> 0:27:51.650  
Speaker 1  
Hmm, OK.

0:27:51.220 --> 0:27:54.970  
Speaker 2  
A lot obliged to follow any specific architectural pattern.

0:28:0.110 --> 0:28:0.290  
Speaker 1  
Yeah.

0:27:54.980 --> 0:28:1.730  
Speaker 2  
Let's say I'm developing a service oriented systems, so that's like a broadly the playground that I'm playing with.

0:28:1.740 --> 0:28:17.220  
Speaker 2  
But then beyond that, it's purely pragmatic way of doing things and of course trying to utilize the architecture principle to the fullest app from a point of view that you're trying to reuse an already established solution.

0:28:17.230 --> 0:28:18.940  
Speaker 2  
So you know the best practice is OK.

0:28:18.950 --> 0:28:22.810  
Speaker 2  
If you want to achieve certain things, you do according to this and that.

0:28:22.820 --> 0:28:25.530  
Speaker 2  
So I work a lot with Amazon Cloud for instance.

0:28:25.940 --> 0:28:32.770  
Speaker 2  
So you follow certain architectural principles, but it's not like you are obliged to, but it's more like an advice.

0:28:32.780 --> 0:28:43.160  
Speaker 2  
So this is something that has been, I'm assessed, proven to help people, but it's not like proven proven, but more like has a good track record of something.

0:28:43.770 --> 0:28:43.970  
Speaker 1  
Yeah.

0:28:43.890 --> 0:28:52.240  
Speaker 2  
So from a that perspective I'm I'm kind of very interested in architecture, but I'm not very bound to follow any specific process when it comes to it.

0:28:52.250 --> 0:29:2.20  
Speaker 2  
So for me it's just to make my life easier and make the life of my team easier so that I can communicate high level ideas much more efficiently with them.

0:29:2.200 --> 0:29:3.620  
Speaker 1  
Yeah, yeah, yeah.

0:29:3.630 --> 0:29:43.970  
Speaker 1  
I mean, no, we we agree and that's why also we we I mean we we did some breeze work on on conformance and yeah we realized that we were tickling perhaps from yeah much stronger point of view that's why we in this work we wanted to introduce the idea of something that is continuous but also as we discuss more I mean a non blocking uh aspect and also like as you said more like OK it's a guideline or something that it's good to have but not a yeah it's it's not a necessity so.

0:29:43.860 --> 0:29:46.570  
Speaker 2  
I mean, I think I can summarize in one sentence.

0:29:46.580 --> 0:30:4.330  
Speaker 2  
If I don't have to to prove anything with respect to marketecture if I don't have to have any argumentation, let's say I'm developing a safety critical system in a very strict we development model when I really have to write requirements, design, solution, refine solutions, start implementing.

0:30:6.250 --> 0:30:7.360  
Speaker 2  
In such a scenario?

0:30:7.370 --> 0:30:12.910  
Speaker 2  
Yes, because you can use tool like this for argument that the certain stages of your development.

0:30:13.330 --> 0:30:33.600  
Speaker 2  
But if you only care about the final executable and everything else is just to make your life easier of how to make that executable good, but you do not necessarily have to demonstrate any quality of any other artifact that leads to your executable, then I would say this might be a bit like a overkill for people to use.

0:30:33.900 --> 0:30:34.80  
Speaker 1  
Yeah.

0:30:33.670 --> 0:30:40.0  
Speaker 2  
It's more like people use architectural blueprints only to make their ideas communicated much more clearly.

0:30:40.10 --> 0:30:46.600  
Speaker 2  
So if I talk to my team, I say OK, we have service oriented architecture, these are the big components inside of the big components.

0:30:46.610 --> 0:30:48.740  
Speaker 2  
We have this more refined components.

0:30:48.750 --> 0:30:51.940  
Speaker 2  
Those will be my modules, classes, whatever.

0:30:52.30 --> 0:30:54.360  
Speaker 2  
So the people can have the picture.

0:30:58.790 --> 0:30:59.820  
Speaker 1  
Yeah, and.

0:30:54.370 --> 0:31:0.150  
Speaker 2  
I can communicate it, but I don't necessarily use to demonstrate the correctness of my solution for more.

0:31:0.160 --> 0:31:5.670  
Speaker 2  
For me, it's like a communication medium rather than a tool to move the solid color.

0:31:3.260 --> 0:31:12.100  
Speaker 1  
I think we we, I mean in a way partially agree, although of course, yeah in the sense that the two things very fast.

0:31:12.110 --> 0:31:29.860  
Speaker 1  
So one thing is definitely as you said it, it's way more valuable or meaningful to look at the in, in, in, in, in at the final product you know of your engineering and in this aspect I think is also we didn't mention.

0:31:29.870 --> 0:31:37.890  
Speaker 1  
But of course this is an initial step of what we are envisioning that is out of out of this.

0:31:43.850 --> 0:31:44.30  
Speaker 2  
Yeah.

0:31:37.900 --> 0:31:48.870  
Speaker 1  
You could get already the code generated not so like once you basically model your architecture, you can get for instance the the skeleton.

0:31:48.880 --> 0:31:53.990  
Speaker 1  
But also you can link behavioral models so anyhow, but this is not quite in place yet.

0:31:54.0 --> 0:32:0.10  
Speaker 1  
We have some code generation, but we didn't want to push in that direction because it's a very basic.

0:32:18.820 --> 0:32:19.30  
Speaker 3  
Yeah.

0:32:19.40 --> 0:32:19.420  
Speaker 3  
Yeah, comma.

0:32:0.100 --> 0:32:22.450  
Speaker 1  
Another thing that it's it is fun that you mentioned is that this summons on thing because one of I mean one one while doing this research of course we we started to look also at already exist in dual and we found for instance that the there is this Amazon composer and or remember the name but it's a tool that basically uh I'm maybe you remember.

0:32:25.260 --> 0:32:25.950  
Speaker 1  
Ah, OK. Yes.

0:32:22.320 --> 0:32:27.80  
Speaker 3  
Yeah, it it is a Amazon composer, Amazon AWS composer.

0:32:27.590 --> 0:32:28.270  
Speaker 3  
This is the right.

0:32:27.650 --> 0:32:30.450  
Speaker 1  
So that allows you to to basically model.

0:32:31.70 --> 0:32:31.340  
Speaker 1  
Umm.

0:32:31.530 --> 0:32:32.890  
Speaker 1  
Uh.

0:32:33.10 --> 0:32:42.400  
Speaker 1  
This uh, let's say application based on, as you say, the best practices or principle and all that are found to be effective at AMA.

0:32:42.610 --> 0:32:49.110  
Speaker 1  
But the the point is that in this what we realize is that and or not what we realize.

0:32:49.120 --> 0:33:0.320  
Speaker 1  
But what I think it's the catch is that while you do this, for instance, uh Adama with Amazon composer or any other tool or with pen and paper.

0:33:1.220 --> 0:33:4.910  
Speaker 1  
Uh, you can still put, let's say, the component.

0:33:5.120 --> 0:33:5.810  
Speaker 1  
You can boot.

0:33:5.820 --> 0:33:6.60  
Speaker 1  
Yeah.

0:33:6.70 --> 0:33:20.570  
Speaker 1  
The cloud provider you can boot, so I don't know the application that reads, but the fact that is that you don't have any any any check over the the the semantic and you can basically define whatever you want.

0:33:20.580 --> 0:33:22.650  
Speaker 1  
So the point is that you can draw it.

0:33:22.660 --> 0:33:28.20  
Speaker 1  
You can have it as like as you said the tool for like brainstorming.

0:33:28.30 --> 0:33:35.840  
Speaker 1  
But then if you have an experience, let's say member of the team that connects something wrong, this tool doesn't tell you.

0:33:35.850 --> 0:33:38.960  
Speaker 1  
OK, this connection is not valid.

0:33:39.130 --> 0:33:48.910  
Speaker 1  
So while what we wanted to yeah, the and what we of course stress in this case is also the fact that this is in a way we see it as a lightweight maybe.

0:33:49.240 --> 0:33:53.490  
Speaker 1  
Hmm, not so much from your perspective, but from our perspective.

0:33:53.500 --> 0:34:4.780  
Speaker 1  
It's like lightweight tool that can also give you beside the power of reasoning and brainstorming, and also give you some sort of insurance that you are going towards a direction.

0:34:14.10 --> 0:34:14.880  
Speaker 2  
Yeah.

0:34:5.360 --> 0:34:15.470  
Speaker 1  
But of course we understand that this is maybe not the main relevant in given context, as you say, where the actual focus is on the end product which is.

0:34:15.470 --> 0:34:29.210  
Speaker 2  
And I think the name of the the the you hit the nail in the head in a sense that for you to be able to derive any meaningful do not meaningful but very strictly then you need to have a very rich semantics.

0:34:29.600 --> 0:34:34.110  
Speaker 2  
So as you say like this tools probably lack semantics at all.

0:34:34.120 --> 0:34:35.590  
Speaker 2  
Like maybe they have something.

0:34:35.600 --> 0:34:35.960  
Speaker 2  
OK.

0:34:35.970 --> 0:34:39.450  
Speaker 2  
These two components when when connected together and blah blah blah blah.

0:34:39.760 --> 0:34:45.290  
Speaker 2  
But this is the final line final line that you need to walk on.

0:34:45.300 --> 0:35:1.270  
Speaker 2  
It's like how much semantics you include in it, which will make it very hard and very exhaustive for people to use in a sense like, I will need to spend a lot of time in doing this architecture versus the what kind of information, what kind of knowledge I can infer in an automated manner.

0:35:1.280 --> 0:35:20.40  
Speaker 2  
So if you have enough semantics so that I can check for some specific properties like information leak or I don't know what else, and finally if I can automatically generate code even if it's skeleton, then the question is how much burden it is for me to build such a model with such a rich semantics.

0:35:20.50 --> 0:35:21.180  
Speaker 2  
That that's my point.

0:35:23.520 --> 0:35:24.730  
Speaker 1  
Acceptance, yeah.

0:35:21.190 --> 0:35:25.300  
Speaker 2  
If it requires me to do a lot, what color do you can I extract?

0:35:25.310 --> 0:35:28.500  
Speaker 2  
So that's like the balance that you need to strike, I would assume.

0:35:28.650 --> 0:35:28.920  
Speaker 1  
Yeah.

0:35:28.930 --> 0:35:42.980  
Speaker 1  
Yeah, and that's exactly what we also we wanted to mitigate in a way by providing this again non blocking process along with the let's say possibility to define your own strictness of.

0:35:45.660 --> 0:35:45.860  
Speaker 2  
Yeah.

0:35:43.50 --> 0:35:59.970  
Speaker 1  
So in in principle, I mean you could also use this tool just for like drawing and ignoring all the the violation you know just to be because you are I mean I know this is that we are maybe a bit over the time and I don't want to keep you lot longer.

0:36:0.60 --> 0:36:3.990  
Speaker 1  
I just wondered for the sake of the yeah, the the the validity of the study.

0:36:4.0 --> 0:36:12.510  
Speaker 1  
I just wanted to discuss very briefly this point that as I said, we did some early pre study, let's call it like this.

0:36:12.520 --> 0:36:26.90  
Speaker 1  
And during this pre study with identified some initially were requirements but then we split it in as you can see in challenges more like yeah let's say challenges and then requirements of the process and of course of the tool.

0:36:26.300 --> 0:37:23.180  
Speaker 1  
So I just wanted to gonna check very fast the now that that you saw that tool and also we discussed a little bit about the process if you could come and again on this challenges and how do you think we we we did in addressing these challenges starting of course with conformance and the challenge here was the lack of formalization of conformance here yeah, we discussed at the beginning of course that we formalized the to come four months in a way that as a distance function that we also use for all the tools the other the other challenge was the evolution meaning that we wanted to have a process and a notion of conformance that could be applicable for evolving architecture not just something that is static.

0:37:23.190 --> 0:37:25.50  
Speaker 1  
OK, you shake it once and that's it.

0:37:25.370 --> 0:37:33.70  
Speaker 1  
Automation, which clearly is very important, and the adaptability which is yeah, it's relates to the this.

0:37:33.360 --> 0:37:35.310  
Speaker 1  
Yeah, uh.

0:37:35.320 --> 0:37:53.320  
Speaker 1  
Threshold or or balance that we were discussing just a few minutes ago on how much uh St the conformance check can be so meaning how much let's say effort you need to put in more than in something versus having something more more generic uh.

0:37:53.660 --> 0:37:58.280  
Speaker 1  
So again, yeah, if you do, do you think that we more or less address this?

0:37:58.360 --> 0:37:59.380  
Speaker 1  
Uh, not at all.

0:37:59.390 --> 0:38:0.850  
Speaker 1  
Or yeah, whatever.

0:38:0.860 --> 0:38:2.820  
Speaker 1  
What is your gut feeling?

0:38:4.640 --> 0:38:6.70  
Speaker 2  
I mean, I saw some formulas.

0:38:6.80 --> 0:38:13.730  
Speaker 2  
I cannot check their correctness, but I would assume see one is addressed because you have it quantified and you provide the formula.

0:38:16.950 --> 0:38:17.130  
Speaker 1  
Yeah.

0:38:13.740 --> 0:38:22.140  
Speaker 2  
I would assume for how to calculate its definite values, so you have a formalization of what conformance is, meaning that you have defined semantics.

0:38:23.630 --> 0:38:32.860  
Speaker 2  
Otherwise you will not be able to derive the the number definition of a conformance process that can support evolving references software architectures.

0:38:33.190 --> 0:38:44.340  
Speaker 2  
This is something that I'm not sure because I thought this is the part where you have the script on the side, but then I in the end assume that script is the C4 when it's the you define.

0:38:45.840 --> 0:38:47.30  
Speaker 1  
Yeah, except yes.

0:38:44.350 --> 0:38:48.820  
Speaker 2  
The strictness of this, but can you argue with the script also addresses.

0:38:48.830 --> 0:38:51.750  
Speaker 2  
See too, that it's a definition of process like in a sense.

0:38:52.640 --> 0:38:57.180  
Speaker 2  
Umm you define what you need to check more strictly or strictly or not.

0:38:57.440 --> 0:38:57.810  
Speaker 1  
Yeah.

0:38:57.820 --> 0:39:5.140  
Speaker 1  
And also this with the way we are try to address this is as as we said for instance that imagine that you have something that is evolving.

0:39:5.150 --> 0:39:11.720  
Speaker 1  
Then again, you may want to have this partial checking so you won't do.

0:39:11.730 --> 0:39:15.910  
Speaker 1  
Maybe check only a part of your reference architecture or something like this.

0:39:15.920 --> 0:39:22.890  
Speaker 1  
So this is also the way we try to address this, this, this challenge, yeah.

0:39:24.790 --> 0:39:27.560  
Speaker 2  
Yan C3 it's uh automation.

0:39:27.570 --> 0:39:30.810  
Speaker 2  
Of course you have an automation that was obvious to me from the video.

0:39:31.160 --> 0:39:31.360  
Speaker 1  
Yeah.

0:39:33.300 --> 0:39:36.570  
Speaker 2  
Umm but I'm not sure what to comment on.

0:39:36.580 --> 0:39:51.110  
Speaker 2  
Manual conformance checks and feasible again, we are going back to the kind of circular argument like how exhaustive, how, what do I need to check like OK, do I need to check that I have following service oriented.

0:39:51.120 --> 0:39:53.270  
Speaker 2  
Do I need to check for a specific components?

0:39:53.280 --> 0:40:4.270  
Speaker 2  
How they're connected and so on so forth depends what kind of level of details I want to check and extent of the architecture of to which I want to check those details.

0:40:4.280 --> 0:40:6.560  
Speaker 2  
Then it might be infeasible, but.

0:40:6.780 --> 0:40:7.460  
Speaker 1  
Yeah.

0:40:7.550 --> 0:40:9.500  
Speaker 1  
And and I mean this is, uh, you are.

0:40:9.510 --> 0:40:15.840  
Speaker 1  
Yeah, you're completely right in the sense that again the the the fact that most manual checks are invisible.

0:40:16.650 --> 0:40:39.700  
Speaker 1  
It's it's are gonna say it's a statement that of course we found in the literature, but starting from the uh, the the assumption that you actually need to check the conformance rather strictly for uh reason such as a standard or.

0:40:39.760 --> 0:40:46.50  
Speaker 1  
Or uh, other other and uh other reasons.

0:40:46.60 --> 0:41:7.300  
Speaker 1  
What I wanted to just, uh also show you a little bit because I mean at the end this is not a secret this for instance when we say I mean relation to what we said about the manual checks being unfeasible, is that even in this case which is a simple case in a way because here you have a reference architecture of variety.

0:41:7.310 --> 0:41:9.190  
Speaker 1  
Yeah, Iotc system general.

0:41:9.200 --> 0:41:10.950  
Speaker 1  
Very my level.

0:41:10.960 --> 0:41:21.510  
Speaker 1  
And then you have here an architecture specific one which is called FireWire, which uh you we can argue it's not very complex because you have maybe around 10 component.

0:41:21.520 --> 0:41:31.100  
Speaker 1  
But again here to manual check whether, for instance, yeah, one of these connection is legal or not with respect to this.

0:41:31.160 --> 0:41:44.200  
Speaker 1  
In case you have to, because again, that's the the assumption it's it's not trivial because yeah, even though there are very few components, you start to have a lot of connections and that's what we refer.

0:41:44.210 --> 0:41:53.710  
Speaker 1  
But I mean we I think we all agree that this is perhaps not always needed in, in, in, in, in this context.

0:41:54.500 --> 0:41:58.30  
Speaker 1  
Uh, so this is was just to to to comment.

0:41:58.880 --> 0:42:0.800  
Speaker 1  
Uh, yeah.

0:42:0.810 --> 0:42:7.460  
Speaker 1  
Anyhow, as I said, I don't want to waste uh or still more time since also you said you were pretty busy.

0:42:7.630 --> 0:42:17.930  
Speaker 1  
So I mean, unless you have other final remarks that you want to add, the thing we can we are we may be happy from the talk.

0:42:19.640 --> 0:42:21.550  
Speaker 2  
No, no, it's from my side.

0:42:21.560 --> 0:42:22.810  
Speaker 2  
It's quite interesting.

0:42:22.820 --> 0:42:28.390  
Speaker 2  
It's a bit of a pity that we don't have more time to discuss this, but I don't know.

0:42:28.400 --> 0:42:30.950  
Speaker 2  
I think you kinda conveyed the message quite nicely.

0:42:30.960 --> 0:42:36.331  
Speaker 2  
I don't know how much I understood how meaningful feedback I gave you, but definitely i.e.

0:42:40.700 --> 0:42:40.820  
Speaker 1  
Yeah.

0:42:37.420 --> 0:42:44.230  
Speaker 2  
To the some points that I made, I see some potential for this, but again it's it's a very specialized thing and.

0:42:44.380 --> 0:42:44.920  
Speaker 1  
Yeah, yeah.

0:42:44.930 --> 0:42:46.950  
Speaker 1  
I mean, uh, of, of, of course.