Now the normal way that we have done our introductions in the past for episodes has been asked the
person we're interviewing to simply introduce themselves, say who they are and what they
do. So Christoph, I will keep the tradition going and ask you who you are and what you do.
So I'm Christoph van Tomme, I'm the CEO of Pranovix, which is a consultancy that specializes
in developer portals. It's a bit complicated because we have a product and we do services,
but you know if you want to find out more about that go and look at our website.
I'm a by-engineer by education, but I've spent all of my career in software. So
there's a lot of that ecosystem and platform thinking and that kind of stuff that's sneaking
in through the back door where I'm trying to apply some of the things that I remember and that I see
in my day-to-day, in my garden and around me in nature, that I try to apply to software systems
and social technical systems that we help to create. Both complex and complicated systems.
Complex, I would say. They're definitely complex.
And Cherryleaf and Pranovix have worked together in the past and you wrote an article on the
Pranovix blog and one of your colleagues suggested it might make a good conversation
topic for the podcast about AI and APIs, which are probably the two hottest topics
in documentation. So it might be good to start by, do you want to summarize what
it was that that article was about? This is already a couple of months ago that we
tried to organize this earlier. We had a bit of a hiccup and then the holidays.
But already a couple of months ago, this was when the hype train was really starting to roll out of
the hype station at unprecedented speeds, let's say. I've never seen a trans takeoff with that
kind of fervor in the software world. And I think what I wanted to do was to challenge a little bit
the immediate instincts that a lot of people had was, okay, so how can we use large language models
to serve our documentation? So we're going to replace our documentation site with a large
language model. There's several things that came into that article that I wrote based on
a lot of thinking I've done about software systems and social technical systems and
what's the difference between an organic complex adaptive system and a software system and what
are they different? How are they different and what are they good at? And now what do we need
to do with this? And I wanted to challenge people, you don't necessarily need to go and install your
own large language model. There's lower hanging fruits and there's other things that this change
will trigger in the ecosystem in our wider society that's probably you can take a stance on and that
you can get ahead of the game by doing some other things. That was the intake for the article.
And I've hinted at some things but I haven't gone into the details yet. So as you say, we've been
talking about having this conversation for a while. Since you've written the article, have your
ideas changed? I think it fundamentally they didn't. Well, they have deepened and like last week
at API Days in London, I was actually talking with a couple of people about how to declare
your interfaces. We started talking about maybe we should have a new standard for this and there's
a new blog post that Anita writes that is actually continuation on this. It's also about AI readiness
but it's like a little bit more tangible already. It is still the same idea about making a clear
separation between this generative and authoritative that is still there. And yeah, just like getting
used to computer systems now operating in different modes than what we're used to.
So we've also been on the same path in looking at this at APIs and thinking oh my,
at AI, generative AI and thinking oh my god, this is going to change things and where do we fit in

and what we've done is has developed a training course, a learning course on this. And you've mentioned something there that was in your article and that was actually one of the questions I had and that was about generative and authoritative information. So do you want to expand on that on how you see the difference between the two, why it's important to distinguish between the two?

So I think the best way that I've found to explain it is imagine you're going to an office building if it's still not all remote and it actually still has an office. You go to the front desk and you talk to the receptionist and you ask them, do you know if John is in? Or yes, I'd like to speak to somebody from support or I'd like to know a little bit more about your products. And the receptionist will say, well, from what I know, I think I've seen John coming in, but I'm not sure who that is, but I think he's in. Or yes, I have some vague ideas about what our product is and they'll say something. But then they'll say, let me look it up for you. And they'll go into their system and they'll use an authoritative system that actually knows, like, yes, John has passed the gates. He hasn't left the office. So he must be inside of the building. And then let me look it up. They'll look up their phone number, they'll give them a call, and they'll come back and say, yes, John is in and he's waiting for you in reception area. This is the difference between generative and authoritative. The generative part is what humans are really good at is like giving a first approximation, kind of exploring a little bit the landscape, providing hooks that you can use to start doing research. But then the authoritative system actually goes and fetches the information and says, yes, I am 100% sure this is true or this is not true. And the really interesting part is that we're used to computer systems doing the authoritative bit. We're not used to them doing the generative bit. This is new. And this is where it gets us completely confused and start believing that this is going to be magical or something. But it is just like with humans, when you hire somebody new to do a job, they need a lot of training to be able to do that job. And they'll need a bunch of systems to support them to actually be able to do that job with precision. And in a really qualitative way, like if you just take a person from the streets and ask them to do something, it's not going to work. And even if you train them, even people that have spent decades researching and developing like researchers, they will still rely on their computer systems to go and validate the things that they're saying. Because we just don't remember all, we don't rely on ourselves, or we can't rely on ourselves, on our memory to be able to do all of these things. We have grown to rely on our systems and our technical systems to support us in those functions. So this is the challenge of hallucination, where if large-language models don't know the answer, they make it up?

Yes. And I think the challenge of hallucination is a labeling problem. Because when you go to receptionist and you ask them, what do you think, they'll come up with an answer. But they'll say, I think that I'm not sure, actually. Now, let me look it up. So that is still something that we need to figure out. How do we make these generative systems give an idea about the reliability for their answering? But it goes beyond that hallucination problem, I think. It's about composite systems that are much more capable of doing things than any of the two systems on their own can do.

So we had this initial thing where everyone was thinking, oh my god, there'll be no need for technical authors, technical writers, there'll be no need for documentation pretty quickly. It's become clear that you need authoritative source content

to get good answers out of an API. How do you see the role of curated, well-written documentation in this landscape where people are using generative AI systems to provide the right information? I can imagine content harvesting systems that are self-correcting and that are validating and so on. But I think that's still quite a way ahead in the future. Also, making sense of a whole or just really selecting the good stuff from the bad stuff, because all of what generative AI is doing is basically generate a bunch of random stuff and then little by little getting better at saying good things purely because there's a human that has been training that system. I wonder if in a large language model context, if you were not going to need even more humans to train those systems than in a pure just declarative documentation

model. And I think the other aspect is things like how do you make sure that your large language system has the right information that is up to date? I think you could imagine a system that is just feeling its way to reality and that's when people start complaining or there must be something wrong. But that's kind of a really terrible way of doing business, where you need to have people complain or give negative feedback to the system to start seeing like, oh, actually, probably there's something wrong in my model and then start adjusting ways to system. So I think probably the experience is going to become much more advanced and much more interesting, but the amount of work to deliver that experience might be equivalent or even more than what we have today. So in the past few weeks, and it really is just the past few weeks, there's been a couple of approaches to this problem that have come out. I don't know if you've had a chance or come across

them yet. One is rag or retrieval augmented generation. And the other is to interface with APIs for getting information. The second one I am familiar with. So the second generation is that you use the large language model purely as a way of generating natural language answers. And what you have is you have a bucket of information, which is the authoritative source. Yes. So somebody asked a question. There's a gateway. It looks for the source content that has the answer that somebody's asked. It then prompts the source content into the prompt that goes to the large language model with the user's question and then lets the large language model find from the source where it is the right answer that and therefore the large language model doesn't use any of its own data sets to find the answer. It only uses the information that it's been given from the user documentation and then it provides the answer. That's one approach that the people are taking. You've got limitations with that in that you've only got so much content you can put into a prompt. But that means it only gives an answer from an authoritative source. And then the other way has been discussions on instead of having a database doing it by APIs, which I think you said you may have come across. Yes. I think this is where the big dream for this technology is the idea of a general artificial intelligence and of an AI agent that interacts instead of you with the world. So it can go and search information and kind of like pre-filter information based on what it knows about you to make certain predictions about the content that you probably would want to engage with and then help you to find your way to the right content. And I think one of the things I suspect is happening and would be really interesting to see is that I suspect that the next feature phones, that's the large mobile phone manufacturers are working on, probably will be able to run LLMs like on the device or I don't know. I imagine. They've run LLMs on a Raspberry Pi. I think it was really terribly slow. I don't know yet how

much computation power you really need. I know that it's horrendous and it's a lot actually.
But I would imagine that this layer, this translation layer will move into the edge.
I think there's a lot of things that can go wrong with this when you centralize it.
Imagine you give chat GPT the right to do API calls against your bank account.
That's not a very appealing proposition, right? Imagine that an AI is starting to interact with
the world through APIs. There's all kinds of ethical problems with that. Who's taking those
actions? Is that chat GPT or is it you that took that action? And now you could imagine,
but if you own the device and if you own the model and it's in your own device and so on,
then it becomes very interesting because then it becomes almost an extension of you as a person
that is able to do certain things out there in the world instead of you with your supervision
and your control. That's where we're going towards. And APIs are the perfect solution
for interacting with the world because that's what we've been using them for,
for programmatic access. And now we have something that's becoming more and more human-like
that we can use as a translation filter between us and that programmatic world. So there's a lot
of fascinating stuff ahead of us. Interesting because I've seen there's some YouTube videos
of people have had installing large language looking locally on their desktop and then
putting all of their information in so they can just ask a question and it'll retrieve,
answer it from the Word files and all of the different things. Well, mainly PDFs rather than
Word files from that. And the processing requirements and size of these databases,
I hadn't even considered a potential for it being on a smartphone. But the banking thing's
interesting because one of the risks at the moment with large language models is what's
called prompt injection attacks where you can do malicious coding and there is at the moment no
defence against those. So if you are giving an AI system access to behind the pin banking,
then there is a risk that somebody could inject code via a prompt in and get into a banking system
and potentially raid one or more persons and accounts. But like the real value of these systems.
So if this is what we're going for, for this layer between humans and machines,
then we will need to make sure that the human that owns the bot is approving what's happening
in their name. We've been touching on this now really of AI agents. Do you want to talk
a bit more about what you wrote and all your current thinking on that aspect?
Platform APIs and developer portals? Yeah, and perhaps even AI agents. So platform APIs,
I think there's two layers of APIs that are roughly speaking and there's always
the dangers to make like binary testifications are always hard. But roughly speaking, I think
on the inside of organizations or organizations are working on platform APIs that are APIs that
are enabling that are enabling people inside of the organization to do complex things faster.
And the benefits that they bring is in that's their bridge between doing things at scale
and scalable infrastructure that you might have created and doing innovation. And what
platform APIs do is or what any piece of a platform does is it makes it possible to do
innovation in a scalable way so that you can really rapidly iterate and make changes and do new
developments without incurring lots and lots of technical depth because you've done it in a way
that you're reusing modular building blocks capabilities that you've built in your organization
that you can build into those new business applications that you're building.
Then there's on the outside that you have the what I call interface APIs. They're
APIs that are exposing some of those capabilities that you have on the inside

to an outside world either as an API product that you're selling or and this is where actually most of these things are because most APIs are not about monetization. There you have APIs that help organizations to build ecosystems and to interface with their ecosystem partners and facilitating those interactions. You can imagine it like like our cells in our body where you have preserved hormonal receptors from cell to cell to cell but depending on the cell it will do different things. And I think that's when you're interfacing up so you have this on the platform level but you also have another layer on the inter-individual level between organizations. You often well sometimes see that with white labeling of products as well where somebody is selling something and it's delivered by somebody else and it's all controlled via the APIs. So if you order something on eBay with delivery you can track where your parcel is but it's the postal service or whoever that's actually delivering that aspect. And there's some interesting stuff because you start mixing inside and outside and you might be like as you said like you might be branding something as a capability that you have in house but actually it's a bolt capability but I think it to some extent these sometimes can be mixing and sometimes you might have an internal capability that becomes something you're selling also on the outside but at the same time like the way that living organisms work is that they have a boundary where they decide is this harmful or is this going to help me and if it's harmful then you close the gates if it's going to help you and you can pull energy out of it or it's more information that helps you to to become more adaptive and and survive better in your environment then it opens the gates and it lets it come in. So say you're an organization that has those types of APIs within your organization in terms of where AI fits into that is it in enabling people to use and apply those APIs without knowing how to code i.e natural language is it that you ask a question and then the AI knows which of all the APIs it needs to pick and and it's like a clever chaining device that makes that makes it for you is it both is it one more than the other how do you see the AI bits assisting where you have those APIs in place? I see it as the interpretation layer that translates between a human that is just saying something and then a machine that is more authoritative about what is doing but that needs certain prompts and like an exploring like what are the prompts that you need to be able to make the system work and like right now we have the the whole prompt engineering thing where humans are learning a new language it's basically a new programming language that's a lot less exact than what we're used to it's a lot more permissive and and vague it actually it's not even deterministic like this AI system like large language models i think they're not well they're on purpose not deterministic you give them the same prompt and you get something different every time and and that's what actually makes it more human or more relatable it's a linguistics driven programming language in some ways the better you know the English language or the grammar structures of English or another language the better prompts you can rise and the better results you get yes what's fascinating is that we've gone from in software engineering it's all about predictability like you don't want a system that's going to do random stuff you want to make sure that like surprises typically are bugs and so when and now we've developed this new technology where surprises are features and and where the ability to delight us with new answers and unexpected

behavior is actually the main feature of this technology which is fascinating because that's that's a whole different types of software engineering than what we're used to yeah yeah it's a fascinating times let's say it's like that let me go back to something again you mentioned about having large scale curated api is ready as and when api systems come in to get to that point what are the challenges that organizations face where they might be today to to have that quantity that robustness of apis they're ready to then apply ai i think right now this is about what is the interaction surface you have as an organization and and i think right now that interaction surface first of all that interaction surface is fractal in nature meaning that like explain make it more concrete how if somebody wants to sell a product to your company how do they find out to whom they should sell it how can they start that interaction a lot of large enterprises have created vendor systems that will guide you through a set of steps that you have to go through to be able to get approval and so on and so on so there's some of that but then there's also the initial request for going through that process how does that work in practice today that's probably the interests of individual people that you need to somehow trigger and probably for sales it will always be a little bit like that although that you start seeing now i don't know how it is for you but for me on linkedin more and more i start seeing automatically generated interactions same on when email it's actually not going as fast as i thought it would have gone so it's it still is somewhat manageable but this barrage of input and attention grabbing there's a tsunami coming and to be able to prepare for that tsunami i think that we need to become deliberate about what interfaces we're what ports are open in our organization and how we allow people to interact with us you see this already happening like in support or you see some really terrible examples of that in support where you know some organizations

don't even have support but i'm sure you have had this experience where you have a problem and you start looking and you have to like phone into a phone number that's a paying number and then they put you on hold for half an hour like basically they're they're inflicting a massive amount of pain so this is like one of the terrible examples another one is a chatbot i had this one i ordered a package online and this was the second time that i were not able to deliver to my address because the the person who was supposed to do delivery said that the address did not exist and so i tried to like explain hey probably maybe there's a delivery guy that you have here in our neighborhoods that's not doing their job or something but like this address really exists so you know please fix it but i ended up talking to a chatbot and and we went like three times around where i'm basically starting from scratch again explaining the same thing over and and oh yeah this is conversational design that's where they're trying to solve everything with a machine and it doesn't work because you need a human to catch the slack and to to to be the the grease in the machine to make sure that all the problems do get solved and people are not getting super frustrated what i'm getting to is these are anti examples hopefully we'll do better in the future hopefully we'll have like supports and and this kind of interaction interfaces that are where there's a good fallback and that actually detects when the fallback is necessary when there actually people can step in but i can imagine that to be able to deal with the barrage of messages that is coming that everybody will have to weapon up and that's or arm up that you it's basically it's an arms race that's to be able to deal with the the ai world's and the barrage of messages that we're going to get that you'll have to have your own ai's because otherwise you just won't be able

to get anything done anymore because there's just so much information flooding you all all over the place well the big developments at the moment within ai are a technical headless videos where the idea is that you can have an avatar present on a video you can write you can get ai to generate the text of what the avatar is going to say so you can have a sausage machine that's generating promotional marketing videos all the time and then put them onto places like linkedin twitter and youtube and you can have people in low-cost countries just generating all of this content and you're saying that it's going to be up to us to filter that i was hoping it would be done by linkedin and youtube and so on that they will weed all this stuff out but you could well be right that it gets past them it gets sent direct by email and by instant messaging and we have to deal with a lot of that content i think it is how do you know if her message is legit or not even a generated message might be legitimate but it really depends on the context and a platform like linkedin they have some of your context of your personal context but primarily filtering is being done on on a platform context level and i think there'll be more and more content that slips through and then having something that knows you through and through and that is allowed to know you through and through because that's the other aspect i i would hope but we'll see because it might be another centralized nightmare but i would hope that this could be a new generation of the open web where our smart devices are becoming you know the filter for things but i'm not sure because there's a equally a big challenge that's like one of the big cloud providers is going to be sitting in as a spider in the spider web and knowing you better than your phone and doing this filtering for you who knows i don't know if people will allow it i don't know if governments will allow it yeah it's going to be an interesting question the lazy approach is to let software do the filtering for you which then means that they control what you are seeing which is a well it's kind of the foshenborg and you know you get it for free but they know everything about you and i don't know it's in the world i have my well most of my career i've spent in the drupal community where it's all about the open web and like how can we make sure that people can own the instruments of content creation so that a small business does not have to depend on a facebook page to be able to be in business i think this could either strengthen the open web or it can weaken it further if the investments are too big although that there's just some really interesting things now with smaller like llms that are still quite powerful and then if you do what like those two patterns we talked about where you don't rely on the llm to be sufficiently trained to be able to provide for the authoritative contents then maybe we can deal with slightly more stupid models that are using some of those other sources to provide the the more intelligent answers yeah it's going to be very fast it's going to be fascinating to see how it all works out so have you come across any organizations that are doing this today or on the right direction that people should keep their eyes and ears focused on so i had a conversation last week where it's a developer at paypal and he said that they just published a graph of all their apis so that's i'll need to go and confirm that i can share this but i'm pretty sure that i can share it but he said that basically what they wanted to do was they wanted to enable AI to consume apis and for that reason they had to publish their graph and that's what he did and he said and there's a couple of other companies that have done this github has one of these i think microsoft has one of these so yeah this publishing your interfaces i think is what the step is that might not be a developer portal that might be something a bit more lightweight

but this this is i think a really really fascinating area and there's already some initial companies that are that are doing this and for companies that want to be ready for AI what advice would you provide in terms of steps that they can take now to to prepare for that that wonderful day so i think first of all start working on your apis and it's a little bit tricky right because if you're a really small company then well well how does that work but i think it is about becoming intentional about your interfaces it's like how do you allow people to interact with you what ports are you going to leave open and what ports are you going to close down and how do you do that in a way that it's actually enables innovation and enables success rather than filters out opportunities and closes you down from opportunities and so if you're really a large organization and you already have apis then you really need to start working on okay how are you going to publish that and how are you going to make it accessible for potentially for llms or for some next generation thing that is going to come after this how do you communicate about those apis and api design and things like that how do you address the skeptics that might i mean within every organization often there's the battle of inertia the the option of doing nothing how would you address concerns people say again the ai is hype or that the apis are not necessarily beyond be all an end all some benefits or drivers for acting so from a certain size of organization having platform apis or having a platform like having an idea about what is your platform that allows you to be adaptive and reactive at scale i think is is inevitable and is essential and what you'll probably see and one of the best places to start building your platform is probably thinking about what capabilities you can abstract behind an api from a certain scale like if you're really small business maybe you don't need this or maybe you can buy some of these capabilities from organizations and even if llms are going to fizzle which is unlikely i would say but it still is possible that's regulation or whatever throws a wrench in this the story or that some of the promises are not fulfilled but even then building a platform and being deliberate about your interfaces with the world is going to pay off no matter what you do because it will help you to be to scale to be at the same time be scalable and resilient now what this is comes from the three economies model from j bloom that i'm sure i'm not doing full honor too because it's probably better to go to the source but he talks about the three economies under which a team can work so there's the economy of scale which is a team that is trying to reduce costs by doing the same thing over and over again and reducing variability there's an economy of differentiation which is like in teams that run under economy of differentiation are creating new variety so that you can capture more value from customers and you can create better products that address better customer needs so you can charge more money for it and then he talks about the third economy which is an economy of scope which is this platform economy which is about creating things that become better from reuse and those are things like apis because apis in truth are just designs they're an api is a contract it's not an api itself is not the server that is running the api the api itself is purely the contract so you you can imagine what probably will happen is that we're going to see a lot more standardization in apis because right now we're in the era of apis a product which means that companies create their own product their own api products that they have to run like a product like a service and you know support or

whatever whatever but i think that we're moving toward a world with apis's utility where standards api designs are reused across the industry and everybody's using the same api designs and this we start already seeing some of this happening like in the banking sector in the telco sector some of the standardization is happening and that's so you could say you could take a sit and wait approach where just wait until other people have done the standardization work because it takes quite some effort at the same time the internal transformation you need to be able to work through apis and to leverage apis that's not simple and and i think it's a high time to start working especially in larger organizations to start working on the capabilities to be able to to take advantage of these technologies yeah there's some consistency isn't there with things like know your customer and identity management across bankers and telecoms where they they generally follow a standard or they are very similar between different apis this is like right now even even the open open banking even the psd2 regulation that was implemented in the banking scene i don't well there's exceptions but almost all banks have implemented their own version they've got different apis it's a mess but i think because today having an api having a good api is still a differentiator but as these things will standardize and become more and more just table stakes i think

that a lot of a big part of the api surface will become more standardized and it will become more like a utility rather than a product where yeah you can you can just say okay i want i want a payment done i don't care about what service i just want this payment done and oh well i want it done well but but i don't care how it works is there any question that i haven't asked that i

think i should have asked who that's an interesting question i've covered a lot maybe

yeah nothing pops up right away there's a lot more service to cover like the the platform thinking ecosystem thinking but i think what is relevant to this conversation to this topic

has has been addressed i would say your so your article on the pronomics website that's on the blog and it was let's have like three ideas on ai readiness the role of apis and developer portals in generative ai systems so people want to know more i guess that would be the starting point would be to look at that post yes there's a sister post that's brewing about how can we create a common standard for declaring interfaces so that well basically organizations can say these are all our interfaces or these are the interfaces we want to share and then make those from then on not only available for humans but also to robots because if you think about it how do people find out that you have a banking app so you're banking with a certain bank how do you find out that there's an app most likely somewhere in the footer there's a link to the the play store or to the iTunes well the apple store yeah but what about more niche applications how do you find out that there's an integration with your favorite billing software probably you have to go to your billing software and hope that they have a marketplace and find your bank in the billing software ace is kind of crazy right so we have sprinkled breadcrumbs are over our digital presences that people need to go and find and be lucky that they find to figure out that there is an interface that you can use to do a certain job which is insane so i think thinking about how can we change that so that we can become more deliberate in declaring what interfaces we accept interactions through and then maybe you can find the back door still but that they becomes it becomes more clear what are the sanctioned channels like the shipping routes

for information where it's safe to go because here you're going to go really fast and you're not going to bump into any sandbanks i think what we need what we need to work on and that that's the article i'm currently thinking about and as i said like i had this conversation with swap nil sapar and a couple of other people at the api days in in london like john from sonofi and this danyek nemeck from superface like there's a couple of people that that we have a conversation about like hey wouldn't it be cool if we if we would be able to declare our interfaces and i do you know just like you have the robot txt or something like that to say this is how you get support here or and here's our app for this capability and there's our app for that capability that's the integration for this system and for the other stuff here's the api this is something like that and what about just regular customers how do they find out what's what your interfaces are i can even imagine that your opinion union is going to regulate this and say every single company needs to have one of these files with at least support at least this and at least that and for support you need to have this kind of fallback in case your machine doesn't do it and then you have to go to a human i can imagine this is going to be regulated so that this kind of games where companies go and hide behind phone paywalls and whatever other really really toxic behavior that that is no longer possible so that kind of thinking this is what i'm working on and it's connected to something i've been chewing on for a long time which is the interface manifesto which is a complexity philosophical perspective on how interfaces help us to be more adaptive but that's really half baked and it needs at least a couple of months if not a couple of years of seasoning before i can come out with that but yeah but this is this is what it's inspired by that yes the well that looks like it'll be an interesting article so that will be coming out shortly i guess and if people want to contact you what was the best way linked in or pretty much it used to be it used to be twitter but yeah thrashfire so linkedin is currently the best way to to connect and to to get in touch there's also our website of course and if you have an inquiry about developer portals then my colleagues or me i will very happy to answer your your contact request if you follow the proper interfaces actually yes this is the same thing yeah same challenge so we've done a lot in that course in this conversation so christoph thank you for your time thank you alice thank you very very much for reaching out and for giving me this opportunity to to have this conversation and it anyway was already way too long since we last talked so it's good to catch up and yeah thank thank you for doing the podcast and for having me as your guest you're welcome