



# Notes

Apr 3, 2025

# Dev Team

Invited Kiyasha Singh Fanyana Nkosi Johannes Backer Justin Germishuys

Attachments 📁 Dev Team

Meeting records 📄 Transcript

## Summary

Justin Germishuys presented updates on several projects: a research agent (using Celery and Firebase to overcome concurrency issues), a video processing pipeline for sales training, and a devil's advocate agent leveraging Bloom's taxonomy for sophisticated argumentation. Johannes Backer raised concerns about Mistral OCR batch processing and Docker image hosting; Justin Germishuys recommended Replit for simpler projects and Google Cloud Run for more complex needs. The team will continue development, focusing on improving GitHub workflow understanding and exploring cheaper LLM alternatives to Cursor and Max.

## Details

- **Research Agent Development and Challenges** Justin Germishuys discussed the development of a research agent, encountering difficulties with concurrency issues, leading to the integration of Celery and Firebase. They are aiming to enable extensive internet research with results comparable to OpenAI's Deep Research, potentially integrating it into various products ([00:00:00](#)). Despite progress, five research tasks assigned 40 minutes prior remained unfinished, indicating unresolved issues ([00:01:15](#)).
- **Video Processing Pipeline for Sales Training** Justin Germishuys announced the need to build a video processing pipeline to analyze sales training videos and generate performance reports ([00:02:44](#)). This is part of an ongoing project with an external partner.

- **Large Language Model (LLM) Selection and Cost Considerations** Justin Germishuys detailed their experiences with various LLMs, expressing frustration over Cursor's decreasing performance and the cost implications of using more capable models like Max ([00:02:44](#)). They explained that providers like Google are making cheaper models less capable to incentivize use of their more expensive options ([00:04:10](#)). They are exploring cheaper alternatives like Deep Seek and Quinn ([00:05:07](#)).
- **Devil's Advocate Agent Development and Functionality** Justin Germishuys presented their devil's advocate agent, designed to challenge positions using Bloom's taxonomy. The agent assesses the user's sophistication level and selects appropriate tactics from a set of 27, explaining its reasoning process ([00:06:12](#)). They also developed a version where two AIs engage in a sophisticated, essay-style dialogue ([00:08:53](#)). The agent determines when to stop based on criteria like depth of insight and a predefined round limit ([00:10:05](#)).
- **Deployment and Hosting Choices** Johannes Backer inquired about hosting Docker images. Justin Germishuys recommended Replet for simpler projects but suggested Google Cloud Run for projects requiring system-level libraries or parallel processing ([00:14:35](#)). They used Google Cloud Run due to prior experience and its integration with Firebase and Cursor's support for GCP deployment ([00:15:44](#)).
- **GitHub Collaboration and Workflow** The team discussed using GitHub for collaboration, with Johannes Backer already setting up a repository and inviting team members ([00:21:17](#)). Justin Germishuys explained the use of branches for code management and the importance of code reviews before merging into the main branch for production updates. They acknowledged a need for improved understanding of GitHub workflows ([00:23:06](#)).
- **Mistral OCR Batch Processing Challenges** Johannes Backer encountered difficulties with Mistral OCR's batch processing API, seeking advice from Justin Germishuys. Justin Germishuys suggested multi-threading as an alternative to Mistral's batch processing, acknowledging the possibility of rate limiting as a potential issue ([00:25:27](#)) ([00:27:25](#)). Johannes Backer noted Mistral's batch processing offered cost savings, but they might not need that feature, given their current usage ([00:28:26](#)).

- **Future of OCR and Data Accessibility** Justin Germishuys speculated that widespread use of OCR by others might reduce the team's need for OCR processing in the future, due to increased availability of text-based documents ([00:28:26](#)).

## Suggested next steps

- ☐ Johannes Backer will explore Mistral's batch processing API for document processing; if unsuccessful, he will implement multi-threading for parallel processing.
- ☐ Fanyana Nkosi will investigate Gemini 2.5.

*You should review Gemini's notes to make sure they're accurate. [Get tips and learn how Gemini takes notes](#)*

*Please provide feedback about using Gemini to take notes in a [short survey](#).*



# Transcript

Apr 3, 2025

# Dev Team - Transcript

00:00:00

**Justin Germishuys:** um a URL for the call back. They need to just be able to pull the response. So another endpoint that you can run to say is this task done or not. But now the the other problem is that when you try to run them concurrently, they all kind of get gummed up. So I ended up having to use celery to send it to a readers temporarily temporary database. It wouldn't write back to it. So I then had to bring in Firebase and now it's this big gigantic Frankenstein monster. And I'm still not even entirely sure whether the thing works. And so that

**Fanyana Nkosi:** built

**Justin Germishuys:** has

**Fanyana Nkosi:** a

**Justin Germishuys:** been

**Fanyana Nkosi:** whole

**Justin Germishuys:** my

**Fanyana Nkosi:** new system. built a whole new system.

**Justin Germishuys:** but if it works then that means that we can um pretty much use open research to do um you know hundred of hundreds of instances of extensive internet research um and then pull back like full research reports with citation same as you would get from open AI's deep research but you you can just run it and build it into various products.

00:01:15

**Justin Germishuys:** So you can imagine like POV being able to go off and do market research before spitting out a result or as I said devil's advocate that can kind of go off do its agent thing come back and then carry on. Uh so that's kind of but I just checked and the five research tasks that I gave it to do almost 40 minutes ago are not done. So, I have absolutely no idea why those failed, but

**Fanyana Nkosi:** Sounds

**Justin Germishuys:** in any case,

**Fanyana Nkosi:** complicated.

**Justin Germishuys:** there we go. But, um, I do wanted to show you, um, sorry, let me just quickly see if I can find this here. Uh, I've been spending way too much time on this as well, but I have a call with um, okay, let's just quickly clear this. Why? Okay. Okay. Something is okay. So, let me just close this and open it again. Okay, so a couple of things that I suppose I just need to put on your radar. Things are moving along with the NT sales training thing.

**00:02:44**

**Justin Germishuys:** They're going to send us a bunch of videos and we're going to have to process those videos to create reports on how the salespeople did. So, it's videos of salespeople selling or giving a presentation. And so we're going to need to build a kind of a pipeline that can process these videos and output reports for how the people did. So that's like the sum total of our part in the project. Just giving you a heads up. Um for me, I'm speaking to Cindy Hoots next week about Socrates and so I'm building the devil's advocate agent out a bit. Sorry, I'm just trying to get this thing to run here. So, I haven't put an interface on it yet, but it's super complicated at this point in time. Um, oh, regarding cursor, I found that it they keep labbotomizing it, which is frustrating. Like, it becomes dumber and dumber and dumber and dumber and dumber and dumber. Um, but then what I've noticed is that um they've included Gemini 2.5 Pro. So I've been switching back and forth and then wherever I need something more, I just switch to Max.

**00:04:10**

**Justin Germishuys:** But if I'm not getting Max to build the whole thing, then it's actually not that it's more expensive for me to keep trying to use a dumb agent that fails over and over and over and then screws up the entire code base in the process. So, it's just false economy to not use Max. Um,

**Fanyana Nkosi:** What?

**Justin Germishuys:** but I have no idea how much um

**Fanyana Nkosi:** Why

**Justin Germishuys:** how

**Fanyana Nkosi:** would

**Justin Germishuys:** many

**Fanyana Nkosi:** they do that? Why

**Justin Germishuys:** Sorry.

**Fanyana Nkosi:** would they Why would Why would they make the agents dumber

**Justin Germishuys:** So,

**Johannes Backer:** cuz

**Justin Germishuys:** what

**Johannes Backer:** now

**Justin Germishuys:** they would

**Johannes Backer:** Max

**Justin Germishuys:** Yeah.

**Fanyana Nkosi:** to

**Justin Germishuys:** Um

**Fanyana Nkosi:** pay

**Justin Germishuys:** Yeah. So that you pay more. So basically what they were doing is they were giving you Clawude um the Clawude thinking version with extensive thinking for the same price as everything else. And so people were slamming it. And so now they put that under max and charge more for it.

**00:05:07**

**Johannes Backer:** I think it's more than that because with Max it has a it's like it's it's not afraid to put context into its context window. Like when I use

**Justin Germishuys:** Yeah.

**Johannes Backer:** Max it read 500 lines of code out of a file. When I don't use max, it would read like 50 to

**Justin Germishuys:** Yeah.

**Johannes Backer:** try and stuff.

**Justin Germishuys:** Yeah. So, I suppose it's a a mixture of those two things. Um, but I do think that soon we're going to get out of this because as I said, I've been trying I've been playing with Deep Seek and Quinn particularly to support the research agent and they're much cheaper and they're faster. Um, but they're not quite there yet, but soon they will be and then it won't matter. then cost will be. So I think they're just going to try and make as much money as they can at the moment. All right. So let me kind of go here um and then run this and I'm going to share my screen.



00:06:12

**Justin Germishuys:** So there'll be a bit of a better interface soon. Um okay. So, what position would you like the devil's advocate to challenge? And I'll say I want to use Bloom's taxonomy for my project. You know, you can put in whatever you want in here.

**Johannes Backer:** I understand.

**Justin Germishuys:** Um, this is something I need to change. So, now what it's doing is it assesses kind of my sophistication level. So, as I have the conversation, depending on how well I'm articulating my position, my vocabulary, the jargon I use, it attempts to assess how sophisticated I am and then adjust its level to how sophisticated I am on this topic. So, if I'm a complete newbie, it's not going to use like jargon laden language. What it does then is I have like six different categories of tactics and then altogether I have about 27 thinking tactics that I've turned into tools. And so each round it says given the topic which category of tactic should I look at and in that category which specific tactic would be most productive to challenge this person.

00:07:42

**Justin Germishuys:** So if you have a look here, it's suggested the questioning category and it has Socratic questioning, definitional challenge, values, exploration, scope boundary, epistemic challenge. And then it thinks a little bit and it says, "All right, so they're all in on Bloom's taxonomy, but I can tell there's some fuzzy thinking there. Questioning feels like the right path to take. Let's dig deeper. I could go Socratic and ask those open-ended questions, but I think a definitional challenge will really make them clarify their terms and concepts first. It'll push them to define what they mean by bloom taxonomy and how it fits their project. Say yeah, I'll get them to break down those key terms and really think about it. And then it selects it and it produces its response. So all of this will basically be hidden from the user. Those are all the thinking steps involved involved in choosing the tactic or tool to use. Um, so yeah, you can see it's trying to use a fairly casual conversational style. Um, so each round it'll attempt to determine which tactics best.

00:08:53

**Justin Germishuys:** So unlike just using open AI where you say please play devil's

advocate and then you just have a conversation where it won't necessarily use specific tactics. This one will force it to do so. Um so this is just one version. The other version that I created essentially has a AI serve as the interlocutor, the person that the devil's advocate is speaking to. And then instead of the devil's advocate only outputting something like this that a human being would be willing to read, it can basically output an essay in response. And then the interlocutor can respond to all of those points in almost an essay style. And then they're basically having a back and forth dialogue but in essence and that is much more sophisticated because you can get to you can get a lot more nuance a lot more depth a lot more complexity by just letting AI talk to AI than having AI talk to a human being.

**Fanyana Nkosi:** with the back on forth. When does it decide to stop playing devil's advocate? And

**00:10:05**

**Justin Germishuys:** So each round it says are we moving this forward or are we plateauing? Um have we reached a useful level of insight? And it has like a bunch of criteria that it judges. And so I set a kind of arbitrary threshold to say look if at least there is this kind of shift in the depth or the insight then you can stop and then ask the user if they wanted to continue but it could essentially keep going with each one pulling in more things into the conversation. So what I've also done is I've just also implemented an arbitrary cutoff to say look either the conversation ends within 25 rounds or it ends at 25 and I can set that to 50 or 100 depending on what we want to do. Um, so yeah, it's it's kind of a lot, but what I'm hoping for is that by being able to run something like this inside Astroenica, they can take any sticky topic and what it will do is it will intentionally use tactics like getting back to first principles or saying, "Hold on, we need to test our fundamental assumptions here."

**00:11:31**

**Justin Germishuys:** or what happens if we reframe this? And what's nice is that in the nonhuman version where you cut the human out completely, you can actually use five or six tactics in parallel. You don't have to do this kind of um sort of series approach to to exploring the issue. And so I think that's the future in terms of exploring the space or arguing about things within the space. There there's another tactic you can use where it

says okay think about concepts that might seem unrelated but from different fields like psychology, biology, um astrophysics and then bring them in and see if you can use them as analogies to move the thinking forward. And in that way, because it it's intentionally placed in its tool belt to use it, it will it will do those things at some point in the conversation. Um, so we'll see how that goes. But this this is not something we're used to working with because now what happens when AI produces 15 war and peace size books on the topic of snails you know who's going to look at that how are you going to make sense of it how you going to decide what to do based on that so we keep finding ourselves being the bottleneck or the rate limiting factor for AI.

**00:13:16**

**Justin Germishuys:** Like we can set up an agentic system that can run for 2 days and pretty much cover all the literature in a particular domain and argue about it from 500 different perspectives and be able to self-criticize and judge whether it's making progress or not. But then there it's done. What are we going to do with it?

**Fanyana Nkosi:** right now.

**Justin Germishuys:** And so that's kind of like the next conceptual puzzle here is how do we get out of AI's way in a way and then still manage to to get value from it and still in some way know what it's doing and have some say over it.

**Fanyana Nkosi:** It's tough.

**Justin Germishuys:** But in any case, so that's that's cool. Uh so everybody's busy with like a whole bunch of things. Um, cool. So, while we have a little bit of time, is there anything you want to discuss? Yeah.

**Johannes Backer:** So, why did you choose Google to host your doctor your Docker images? Because I that's my next step. I'm going to get over myself, you know, deploy it, see if I can run it locally, but then I need to deploy it somewhere and replet obviously is not not where it's at for Docker.

**00:14:35**

**Justin Germishuys:** Um, look, I mean, sometimes Replet will be able to host your thing without you needing to bother with Docker, in which case go with Replet. Um, so for instance, I can get a version of my um, Open Deep agents running on Replet. But what I cannot do is set it up for hectic parallel processes. So I can't set it up with a readers

database um for instance it just doesn't want to work with that. So it has limitations but often replet is simpler if you don't want to bother with all of that. The only time that Docker is really useful and in deploying to the cloud as opposed to Replet for me is when there are libraries or packages that are system level and you cannot install on Replet

**Johannes Backer:** All right.

**Justin Germishuys:** because they're just completely unsupported.

**Johannes Backer:** Thank

**Justin Germishuys:** So

**Johannes Backer:** you.

**Justin Germishuys:** for example, um Tesseract is one of those things. Replet can't handle Tesseract.

**Johannes Backer:** Yeah,

**Justin Germishuys:** And

**Johannes Backer:** that's

**Justin Germishuys:** so the reason I went with Google Cloud Run in this particular instance was arbitrary.

**00:15:44**

**Justin Germishuys:** Um I had done a project previously where I set up something on Firebase and Firebase is connected to Google Cloud and there's great integration between Firebase and the broader Google Cloud inter ecosystem. So I deployed a thing on Google Cloud Run and so because I had done it, I just did it again.

**Johannes Backer:** Okay,

**Fanyana Nkosi:** Yeah.

**Justin Germishuys:** Um, and

**Johannes Backer:** that's it.

**Justin Germishuys:** sometimes you just try something and it works and you're like, okay, if it's not broke, why fix it? And the other thing is that cursor, particularly when you put in Gemini, kind of understands how to get things running on GCP.

**Johannes Backer:** Oh

**Justin Germishuys:** And

**Johannes Backer:** s\*\*\*.

**Justin Germishuys:** so then right there in cursor it'll

**Johannes Backer:** Okay.

**Justin Germishuys:** tell you okay just log in to Google and then you click a button you

log in it immediately is connected to your um Google cloud it'll create a project for you from inside cursor and once that project then it will know what your project ID is because you created it inside cursor and then it'll just create everything related to that project from inside cursor.

**00:16:55**

**Justin Germishuys:** So, it was just super easy

**Johannes Backer:** That

**Justin Germishuys:** from

**Johannes Backer:** makes

**Justin Germishuys:** that

**Johannes Backer:** sense.

**Justin Germishuys:** perspective. Um, but then of course, like with all of these things, if you don't actually know how to do it yourself or why things are happening, you lose like four or five days in a loop,

**Johannes Backer:** Okay.

**Justin Germishuys:** but you eventually kind of get a get the hang of what's happening like the different steps in the process, the different settings you have for um your deployment, those kinds of things.

**Johannes Backer:** Okay, I'll test it out.

**Justin Germishuys:** Yeah.

**Fanyana Nkosi:** Let

**Justin Germishuys:** So,

**Fanyana Nkosi:** Google

**Justin Germishuys:** sorry.

**Fanyana Nkosi:** I see a lot of people after Johannes told me a lot of people are playing with the 2.5 Gemini. Uh, I have to take I have to dive deep into that.

**Justin Germishuys:** I mean, it currently it's supposedly the best of the thinking models. Um but past a certain point it doesn't really make that much difference. Its big value for me is its context window like the fact that you can it can un you could put your entire code base in it and it can reason over the whole codebase and that's kind of where it's useful from a a coding support perspective I think.

00:18:19

**Justin Germishuys:** Um, but I think that where I'm starting to explore more is um, Quinn and Deep Seek and this the smaller models that claim to achieve the same level of performance as like 01 um because they're they're going to be much cheaper and we'll be able to host them ourselves. So, I bought a bunch of credits on Together AI and uh started playing around with those models there.

**Johannes Backer:** Is that a different platform together?

**Justin Germishuys:** Together AI. Yeah. Um the reason I started playing around with together AI is because this thing there's this package called light LLM and it's basically like uh a model manager. So,

**Johannes Backer:** Nice.

**Justin Germishuys:** um, it it creates kind of like a standard way to to pull in different models into your program or your code. I'm not recommending it specifically. It's just something I was playing around with because it's quite a pain in the ass. you you have like the syntax for a 40 model call and then you have something different for 01 cuz 40 has a system message but 01 only has a developer message and uh GPT40 has temperature settings 01 doesn't allow you to do temperature settings but it does have reasoning effort medium high low and 1 outputs its re it doesn't output its reasoning that's invisible.

00:20:03

**Justin Germishuys:** Claude on the other hand has its thinking under a specific JSON value that you call but when you're using something like deep think or quen the thinking is in XML tags in the response. So they're all different. And when it comes to like building something that uses the output from those things, it becomes kind of a nightmare to try and create different ones for each of them. So it

**Johannes Backer:** Okay.

**Justin Germishuys:** makes sense to kind of have one standard function where you say I'm using 01. It detects you're using 01 and it won't it'll only res look for the arguments relevant to it. Now, of

**Johannes Backer:** Okay.

**Justin Germishuys:** course, you can just go into cursor and build something like that yourself. You just say like here, here are all the models I want to use. Build me a function that allows me to run all of these by detecting based on the model name what to do.

**Johannes Backer:** Okay. I'll go check it out. Sounds pretty cool.

**Justin Germishuys:** Yeah. Um, oh yes, Jiannis, I wanted to ask you, have you made any progress on the MCP front?

**00:21:17**

**Johannes Backer:** I put it to bed. I'm just using the GitHub MCP because I've been also trying to get used to GitHub.

**Justin Germishuys:** Yeah.

**Johannes Backer:** Still still a long

**Justin Germishuys:** Uh,

**Johannes Backer:** way professional

**Justin Germishuys:** speaking

**Johannes Backer:** than that.

**Justin Germishuys:** of which, um, are you have you been able to set up a a GitHub repo for us?

**Johannes Backer:** Yeah. But I had a few questions. So if so is the idea that we all have to use the same email the dev at stride shift.

**Justin Germishuys:** No. No. So, once you've got a repo set up, you can invite team members. So

**Johannes Backer:** Yeah,

**Justin Germishuys:** then

**Johannes Backer:** I already

**Justin Germishuys:** you can

**Johannes Backer:** invited these two to a repo like

**Justin Germishuys:** Yeah.

**Johannes Backer:** a starter repo that I've been working on.

**Justin Germishuys:** Yeah. So basically the idea is that uh if we're working our projects are going to start getting bigger and more ambitious and it's not going to be something that just one of us works on.

**Kiyasha Singh:** Hallelujah.

**00:22:02**

**Justin Germishuys:** So um also it's going to become important for us to be able to share solutions with one another without having to jump through hoops. So that's kind of the

idea is that um we get like GitHub access tokens, permissions, those kinds of things and then when we build things wherever we can just simply push to

**Johannes Backer:** Yeah. So, we had a meeting, the three of us had a meeting earlier this week and then we kind of just get got started with GitHub. Uh, so they created their GitHub accounts with the Stride Shift email and then

**Justin Germishuys:** Yeah.

**Johannes Backer:** I showed you how to go into your GitHub, get the token needed for the GitHub MCP. Um,

**Justin Germishuys:** Yeah.

**Johannes Backer:** just to start, you know, just to start using GitHub because it's a kind of a learning curve. I still don't know when to push. I'm just pushing as much as I can.

**Justin Germishuys:** So whenever you make a change actually so whenever you change something you can push it to a branch.

**Johannes Backer:** Yeah.

00:23:06

**Justin Germishuys:** So a branch is like um a holding place and then typically what we need is some sort of internal process where we get together we do like a code review to say look we think this works let's push it to the main branch and if we do have live um

**Johannes Backer:** Heat.

**Justin Germishuys:** code deployed somewhere we can actually usually set it up on the hosting environment to automatically update the production version of the app when we push something to the main branch.

**Johannes Backer:** H

**Justin Germishuys:** when we merge something with the main main branch. So it'll automatically detect update the production version and then your end user will experience the the updates.

**Johannes Backer:** makes a lot of sense.

**Justin Germishuys:** Yeah. So, we need to probably all kind of get better at thinking about how we use branches and the main um cuz admittedly like I'm not a dev so I haven't worked in a place where I've had to specifically use GitHub except maybe a little bit in my previous company. I had GitHub access and I had to pull things from it but rarely um but never had to really think about the hygiene around it.



00:24:31

**Justin Germishuys:** So I

**Fanyana Nkosi:** Yeah.

**Justin Germishuys:** think we all kind of have to get the hang of that. But I mean once we get the hang of it um whoever comes after us will just benefit from the structures and systems we have in place. So we're just blazing the trail.

**Fanyana Nkosi:** Yeah.

**Justin Germishuys:** But okay, cool. I think then um I just wanted to have a check in and find out what you're doing and maybe just chat to you about things. Um if you do get stuck or if there's anything you need to know or chat about, do let me know and then I'll see if I can find out find some time. Hopefully my time will be freeing up a little bit more soon.

**Johannes Backer:** There is one thing that I've been stuck on. It just

**Justin Germishuys:** Yeah.

**Johannes Backer:** took me two hours yesterday and then I just decided it wasn't worth it. But for

**Justin Germishuys:** Okay.

**Johannes Backer:** this API, the batch processing of documents, I got the boiler plate code from a Jupyter notebook on the website, but for some reason it just doesn't want to work.

00:25:27

**Johannes Backer:** And I want to know, did you have any problems with the batch processing or is it just me that has to bang my head against the table a little bit more? Yeah.

**Justin Germishuys:** So, I never tried batch processing um because I didn't I didn't have that use case. So

**Johannes Backer:** Okay.

**Justin Germishuys:** it's possible. I mean the

**Kiyasha Singh:** I got

**Justin Germishuys:** question is how do you want to process batches? Do you want to put them all in a queue and say process them sequentially and then output something or do you want one function and then you turn it into an async function and you use multi-threading and then essentially what it will do is it'll use like a an await function and gather all the outputs from them. So essentially what it will do is it'll run your mistral

OCR function each one in a different thread in your CPU. And so at the end of that you have the await gather command and that will pull all of them together and return all of them to you.

**00:26:39**

**Johannes Backer:** Yeah. So that is one way. But also I think so when I was reading through the documentation, Mistro has this ability that you can actually upload a folder of documents through the API and then it does all of the parallel processing already. And apparently it's like cheaper. you get some discounts with batch processing. So I can do multi-threading and do them all at the same time with one API key. But I wanted to experiment with this cheaper version for if I ever

**Justin Germishuys:** Oh,

**Johannes Backer:** want to build

**Justin Germishuys:** could

**Johannes Backer:** tender

**Justin Germishuys:** it

**Johannes Backer:** or something.

**Justin Germishuys:** Okay. So, could it be rate limiting?

**Johannes Backer:** Yeah, I have no idea. It's just it fails and the error messages are poor. So I don't know exactly what

**Justin Germishuys:** Yeah.

**Johannes Backer:** but it works without it and it's not that slow. It's like one document every one and a half seconds. So for small use cases, I'm

**Justin Germishuys:** Yeah.

**Johannes Backer:** fine.

**00:27:25**

**Justin Germishuys:** I mean, and if you don't want to use their batch processing, and it's not that expensive, actually, right?

**Johannes Backer:** It's \$1 per thousand pages.

**Justin Germishuys:** Yeah. So, you're you're fine. So, I would say if you can't get their batch to work, just multi-thread it

**Johannes Backer:** Yeah, that's actually very valid. I'll

**Justin Germishuys:** and

**Johannes Backer:** just

**Justin Germishuys:** then

**Johannes Backer:** do

**Justin Germishuys:** you

**Johannes Backer:** that.

**Justin Germishuys:** then you can do like 10, 20, 30 documents together. Um, and that that's fine provided they're not rate limiting your API calls because that's the other downside. It's not like your it's not like the um the service is hosted locally. You're having to send it to them. And if they have a limit of say 10 requests per minute, well then that's all you can do.

**Johannes Backer:** Yeah. So, I just have to check what they rate. I think I don't know what I know it's something like they they have weird rate limiting like per month you can do a certain amount of requests or a certain amount of tokens.

**00:28:26**

**Johannes Backer:** I don't know what it is for PDF processing but I'll check.

**Justin Germishuys:** Yeah.

**Johannes Backer:** Technically it's it's cheap enough that we don't need a use case where we're going to be processing 10,000 documents per month. So

**Justin Germishuys:** Yeah. Yeah. We'll probably hit a ceiling where we've processed all the documents we really need to process and it'll just be like five 10 at a time.

**Johannes Backer:** yeah. Yeah, that makes a lot of sense.

**Justin Germishuys:** Um

**Johannes Backer:** Okay.

**Justin Germishuys:** also I with also think about how the world changes with OCR, right? So imagine everybody takes their um image-based PDF and runs it through OCR themselves and they have an OCR version that they then upload to the internet or wherever then you won't need OCR for that document in the future. Um, so if other people build this into their things before it gets on gets wherever, then that also helps ease our our requirements on that front because

**Johannes Backer:** there.

**Justin Germishuys:** everything should be readable. But okay, let's let's stop there. Um, there's quite a lot that I need to get back to today and I'm sure all of you too. And Kesh, I look forward to seeing what you've you've done by Monday. And good luck with everything, and good luck with tomorrow's lesson plan.

**Kiyasha Singh:** Thank

**Fanyana Nkosi:** Thank

**Kiyasha Singh:** you.

**Fanyana Nkosi:** you.

**Justin Germishuys:** Okay, cheers.

**Johannes Backer:** You okay. Just

**Transcription ended after 00:30:06**

*This editable transcript was computer generated and might contain errors. People can also change the text after it was created.*