**Understanding esrp system.-20230127\_223142-Meeting Recording**

January 27, 2023, 5:01PM

23m 29s

 **Rosco McMahan** 0:05  
Yeah.  
So I know there's a decent amount of frustration and understanding how the system works, so ESRP was designed. There's new ESRP system was designed in such a way that if a client and epub is considered a client.  
Has any issues in their system, they do not need to JIT in order to figure out what's happening or what's wrong.  
The UI is supposed to provide all that information, even if failures or even if it just entered the system and nothing else. The UI still has it if it's in the Cosmos database, it has it.  
I'm.  
So we're essentially trying to treat ESRP as like a a third party kind of thing where you're not, you know, like, you can't actually go to a different system and debug their system like MOO or DCAT or something like that. So we're we wanted to try and do the same style, but obviously if something's missing, then clearly we have to fix it and then add it to the UI.  
No click and drag.

 **Bhaskar Verma** 1:09  
So let's go through in the 1st place, why did we need a new ESRP system like?

 **Rosco McMahan** 1:14  
Yeah, that's a great question. That's a great question. So what was happening is that the old ESRP system, the shared services, one you may have seen it if you've ever poked into the shared service code.

 **Bhaskar Verma** 1:25  
Yes, here.

 **Rosco McMahan** 1:27  
OK.  
That system.  
Is uh.  
Not well optimized in a lot of different ways.  
Uh, there's a giant list somewhere.  
Umm.  
It may actually be in a notes actually.  
So there's.  
Instability of the program isn't really all that great. Yeah, when it sometimes it may just die because it takes up depending on the submissions to shared service, it may just take up a lot of cores or a lot of IO, and then the node just goes down and then it needs to be restarted.  
It uses esrp.exe, so it uses an actual executable in order to performance functions.  
And that requires corpnet access.  
Which is not something that PME supports. So we had, we would have had to like rewrite it ported or something. Anyway just for that.  
It isn't portable.  
Alright, so that if we wanted to onboard new clients, I know Dina, for example was trying to get movie 6 ESD SUVP a whole bunch of other places trying to get it used it. It was kind of a messy situation.  
Uh, it's not very debuggable.  
I don't know if you've had a chance to look at the code, but it it honestly isn't all that great to figure out what's trying to happen on which processor flows or trying to go to where and.  
Since it uses ESRP executable which Jasons is trying to submit, that sort of thing.  
The IT is prone to traffic spikes and it doesn't handle them very well.  
Uh, yeah, when we do a security release on the B releases.  
That's a lot of stuff. So that's like 20,000 files plus.  
Plus, media is like 150 + a whole bunch of other stuff and it's.  
It it can die.  
I mentioned earlier I storage.  
Uh, it's because we use ESRP executable. It writes output to a file and then we have to read from that file.  
Umm. And there's a high possibility during high traffic spikes that.  
We could run out of I of storage as well.  
Since your question.

 **Bhaskar Verma** 3:51  
Yeah, they're answering my question.

 **Rosco McMahan** 3:52  
OK, I'll I'll also send the Visio as well as part of this.

 **Bhaskar Verma** 3:57  
Yeah, sure.

 **Rosco McMahan** 3:59  
Umm.  
So as part of the new design, the new design changes things only slightly, but it does. It does shake up some core concepts that a lot of people may be used to.  
Umm, so in this case.  
What happens is that I'm using two pub here as an example, but you can Switch 2 pub with me pub if it helps.  
So two poem and me pub have their own logic apps. Me Pub has their own ESRP logic app.  
And there's obviously also a letter of the a lot of other logic apps as well.  
Are you familiar with the late flow?

 **Bhaskar Verma** 4:43  
Uh, yeah, Rosco, yes.

 **Rosco McMahan** 4:45  
Excellent. I I'll reiterate a little bit just so that I guess people watching recording can understand.  
Uh, so that a lot of the flow for the ESRP. Ohh and please correct me too to make sure that I'm not lying or crazy about something.  
A lot of the flow used to be that it would do a circular pole on stuff, so it would send it to ESRP and then just wait every five something seconds or 10 seconds or something and then pull the ESRP to see if it was done.  
Alright, So what we've done is we've changed it. We've changed that process to instead of pulling ESRP, if pulls the release document.  
Uh.  
And sorry, I was thinking about RPM's. We'll get to those.  
So when it's polling for the release document, it really just pulls the the Cosmos document field to see if it's finished.  
Now, there's probably a clear understanding of where does that feel get set where? How do things get changed?  
So let's jump into that. When we do a.  
Flow into ESRP. We call it only once and we call the.  
The users or the clients ESRP manager service.  
In this case, it's me. Pub me. Pub has ESRP File Manager SVC or something like that.

 **Bhaskar Verma** 6:13  
Yeah.

 **Rosco McMahan** 6:13  
And then what happened? Yeah, thank you.  
I'm sorry to interrupt. What did you have more?

 **Bhaskar Verma** 6:19  
Yeah. So I've seen this file and it, uh, if the file is already, if there's a link in the release proposal meter document regarding the BLOB URL, I think it does not do anything. Otherwise it uploads to a BLOB BLOB storage and then sends that particular link to to actual one property survey.

 **Rosco McMahan** 6:20  
No.  
Umm.  
Yes. Yeah.  
So what what you've described is, is this flow down here?  
Umm so if if the file is, if the files that are being processed. So in in me pubs processing scenarios there's the release document and then it has for every file it's an RCMD release proposal meditated document.  
And it's I think it's one to one. So every single file has their own release proposal metadata document.  
If any of those files are corpnet specific.  
It calls an external service to say. Can you take this file and upload it to Azure for me?  
And that's where that's where it puts it here. And it's me pub's own storage.  
In that regard, and I think that's, I think we're on the same page about that. Is that about right?

 **Bhaskar Verma** 7:27  
Yeah, correct. Correct.

 **Rosco McMahan** 7:28  
OK, if there's a VSTS is involved, artifact service or or I believe.  
There's a different BLOB storage from HTC. We may already have permissions for that and you may just skip those entirely and just write that information.  
Uh, simply because we don't need to upload them again because they're already in Azure.  
Uh.  
So what it does? So this is the process of trying to submit stuff for release scanning. There's a bunch of metadata information that it that one pub ESRP needs in order to come back successfully and say, hey, I've I've done my job on your documents.  
Uh, those sorts of things are called callbacks.  
You see that this arrow has two sides that start and end.  
There's three different callbacks. There's the success callback.  
So when one pub ESRP is done, it calls back into your ME pub code.  
Uh. And then there is retry and abort.  
When those callbacks are needed is when the retry manager is used. Mepub doesn't use the retry manager as part of their scenario for overall stuff.  
So this was a first sent that enabling it for them.  
Normally what happens is that hey, you've got an ICM.  
And it looks retrievable. So you go to the retry UI and then you just hit retry and it's done. The retry UI is common across all teams, so UPS uses the same UI and me pub uses the same UI.  
As app all the other teams use it.  
I don't think me Puppet is actively submitting things to it with the exception of 1 pub ESRP.  
So that is that is the thing. If you're not familiar with that, I can provide the link to that as well.  
But that only happened, yeah.

 **Bhaskar Verma** 9:26  
Rosco, are we? Why are we calling it three pub ESRP?

 **Rosco McMahan** 9:31  
It's a joke.  
So.

 **Bhaskar Verma** 9:34  
OK.

 **Rosco McMahan** 9:35  
Umm shashank's. Org is A1. Pub is at at a hole and then at some point there was a redesign so you may see things called one pub G.

 **Bhaskar Verma** 9:46  
Yeah.

 **Rosco McMahan** 9:48  
Now I've taken a liking to call. It's not really next Gen anymore because that was developed years and years ago.  
So I've taken a liking to calling it tupo.  
Because it's the second iteration.

 **Bhaskar Verma** 10:00  
OK.

 **Rosco McMahan** 10:01  
So this this would be 3 pub because it's a third iteration of architecture.  
So it's. I'm sorry, it's just a bad joke.

 **Bhaskar Verma** 10:09  
Got it, got it.

 **Rosco McMahan** 10:13  
But this is it's intended to also say that it's us it's it's Arius or piece system. So there's that too.  
So we.  
So you fill in when communicating with the. Yes, the one pub ESRP system. You fill in information in regards to.  
Uh, the callbacks, which is where, where it'll get back into your system as well as the user something called user correlation ID.  
And the user correlation ID is how your me pub service takes what you submitted and then it it makes it's a translation between transaction ID and release document.  
That's all it is.

 **Bhaskar Verma** 10:55  
OK.

 **Rosco McMahan** 10:58  
So if that's not, yeah.

 **Bhaskar Verma** 10:58  
What is the? What is the transition document slash transaction ID?

 **Rosco McMahan** 11:03  
So that the transaction ID is anytime you send a post request to one pub ESRP.

 **Bhaskar Verma** 11:10  
OK.

 **Rosco McMahan** 11:10  
That's one transaction.  
So yeah.

 **Bhaskar Verma** 11:13  
OK.

 **Rosco McMahan** 11:17  
Now there's.

 **Bhaskar Verma** 11:17  
But.

 **Rosco McMahan** 11:19  
Now I think you might be asking, uh, another there's it sounds like there's a read between the lines. Question.  
An operation ID is the other question.

 **Bhaskar Verma** 11:28  
Correct.

 **Rosco McMahan** 11:28  
UH-1 transaction can have multiple operations and operation is for ESRP itself.  
Here.  
So I we batch things based on certain scenarios.  
So if you submit one transaction with 20,000 files, then what we do is we batch that up into 100.  
Which means that there's going to be 200 batches.  
Of 100 files.  
And that's going to be 200 operation ID.

 **Bhaskar Verma** 12:04  
OK.

 **Rosco McMahan** 12:05  
For one transaction.

 **Bhaskar Verma** 12:08  
Project.

 **Rosco McMahan** 12:08  
On.  
The part numbers get into an interesting territory because we don't wanna do cross pollination with part numbers, so we separate that. Each part number gets their own operation.

 **Bhaskar Verma** 12:22  
OK.  
And how is this the orchestrated in one pub ESRP like is it?  
Ohh I was checking that there's something called a durable task framework that is used there.

 **Rosco McMahan** 12:36  
Yes. So yeah, I'll get to there in a in a minute. I do wanna finish mentioning the user correlation ID if that's OK.

 **Bhaskar Verma** 12:37  
And.  
Yeah.

 **Rosco McMahan** 12:48  
So that the user correlation ID is a translation between the clients documents and the transaction ID.  
So for UUP ohh that's a content document for me pub. That's a release document and I believe there's enough information in.  
In a in a user correlation ID to also mention which because I there's hashes that are also returned.  
Uh file hashes specifically.  
Which?  
RCMD's release proposed a metadata documents need to be updated as well, so if you get a a batch back then you should be able to figure out I I need to update this release document with these RPM's and then figure that out.  
Uh, and there's definitely code that's preexisting in in the ME pub ESRP manager is that does that already? So it can be examined in an event of like. I'm not like you're not sure.  
So when mepub, when the one pub ESRP system receives a request.  
The very, very first thing it does is save the request and this is in the event that it goes down or dies or something else.  
Needs to be rebooted. Yeah, some.  
Uh, that way it just it can pick something up and reprocess it no problem.  
It uses as you mentioned earlier, it uses durable task framework as it's back end processing.  
Uh, so it sets up an orchestration and durable task with different layers.  
So the.  
That the initial layer is processing the input, so there's a lot of validation based on what you sent.  
Uh, so did you send? Did you send files and they files missing data?  
Are they valid? Is.

 **Bhaskar Verma** 14:43  
It will give you for you.

 **Rosco McMahan** 14:45  
Is there?  
It like if you specify part number did you not specify with other thing, so it's just really minor validation to make sure that it when we do get deep into processing it's not gonna fail. So it's it's a fast fail scenario. So that's I believe that's layer one and then layer two is processing the files themselves.  
So we pick up all the files, we make sure they if you did not provide a hash or if you provided a different type of hash.  
And then we make sure that the the hash is present or calculate the hash ourselves.  
Uh.  
I I believe that's a file processor level.  
And then the third level is submitting it to ESRP.  
So we take all that information, we generate a SAS Uri for everything and then we just toss it in ESRP system.  
And then the fourth layer is doing a check status on ESRP. So we loop here for a decent amount of time seeing hey, is ESRP done ZSP done. Is it done yet? That sort of thing?  
If it is great.  
Then we go on to the fifth layer if it.  
And when I say if it is, I mean is it terminal, so there's different types of terminal states.  
Uh, failures at terminal state success as a terminal state pending analysis is a terminal state.  
No, it's not. Pending analysis is not a terminal state.  
Umm.  
But in the process of success or failure, it goes on to the last stage which is callback. So it it takes that information that came from esrp.  
Caches it if necessary and then sends it off to your stuff. It sends it off to your me pub SVC SVC.  
Code.  
And it it sends the user correlation ID it sends using the the callback you provided, it sends the user correlation ID.  
Uh, so you know which release document it was talking about?  
And then from there it's supposed to save all that information.  
Into your release documents European's and then it goes back and then when you do that automatically the logic app picks up that the document was certified and then moves on with its life.

 **Bhaskar Verma** 17:14  
Yeah, that makes sense. Makes sense. Makes sense. Uh, so in case of any sort of, uh, terms of state, be it failed or potential malware found or succeeded. So you'll send or the one pub ESRP system will be sending the callback URL to our system.

 **Rosco McMahan** 17:14  
That makes sense.  
OK.  
So the potential pending analysis potential malware is not a terminal state, so it will continue to loop and check status on it.  
Uh, but it will make an ICM.

 **Bhaskar Verma** 17:46  
OK.

 **Rosco McMahan** 17:46  
So that that way it's like a hey, there's something wrong kind of thing.  
I force it. I think it actually only calls back in success for failures it what it does is it calls retry manager instead so it it calls.  
The other callback URL that you gave.

 **Bhaskar Verma** 18:04  
OK.  
Got it.  
Umm so why? Why are we like filing an ICM and that and after after finding an ICM we found find that that particular file maybe took longer than ohh general files take and the the ESRP status was.  
Ending analysis, but we still file an ICM or why is that?

 **Rosco McMahan** 18:28  
Yeah, pending analysis is an interesting state is.  
Umm it only happens when potential malware is found.  
Uh. And then what happens is that either your team or the ESRP team goes in, does some manual investigations, takes a look, and if everything is fine, then they the ESRP team manually switches it to success.  
If they're, if it's stuff is not fine, then they manually switch it to failure.  
But either way, it's it's on the ESRP team itself to actually change that state.  
Uh. When the state is changed, then our ESRP service picks it up and then goes on from there.  
So it's OK if it takes time. I think it we we do that loop for 30 days.

 **Bhaskar Verma** 19:15  
OK.  
OK. And whenever you like, whenever the one pub service system gets the request, where do we save it? Like or or after saving do we update it?

 **Rosco McMahan** 19:28  
The request in its raw form it saved in Azure storage.  
And I think that has a lifetime, so it gets deleted after a little while.  
I now the operation that that gets transformed is in Cosmos.  
Uh, but all the cosmos data gets.  
There's a UI on top of Cosmos, and you've seen that.

 **Bhaskar Verma** 19:52  
Yeah.

 **Rosco McMahan** 19:52  
Uh, so anything in Cosmos would automatically be refreshed under the dashboard.

 **Bhaskar Verma** 19:59  
OK.  
OK.  
And how how? For how long does this data stays in the cosmos DB? Or do we delete some other one?

 **Rosco McMahan** 20:09  
It stays indefinitely.

 **Bhaskar Verma** 20:12  
OK, OK.

 **Rosco McMahan** 20:14  
Uh, and less. You're talking about Dev and I think in Dev we set a timer for us like a month or something like that.

 **Bhaskar Verma** 20:14  
But.

 **Rosco McMahan** 20:22  
It just you know, so that you don't get weird data in there for too long.

 **Bhaskar Verma** 20:28  
So from my understanding that this ESRP new ESRP system is just a a sort of an orchestrator that orchestrates all these operations from a transaction and then does further and just sends the call back whenever these operations for a particular transactions are done.

 **Rosco McMahan** 20:47  
That's correct.

 **Bhaskar Verma** 20:48  
OK. And for each, so let's let's say I send a request from me website to the ERP system and I I create transaction ID for for that particular transaction at the esrp end. Now for let's say I wanted to.  
Oh, oh, get 10 files or let's say 20 files scanned. Then I'll batch them up, and depending on the those batches, I'll create an operation ID based on each of that batch and.  
Uh, and I'll monitor them once they are in pending on once they're in success or or some other in terminal sheets, then I'll send a call back back to the.  
Playing.  
Of which men my case would be meatball, right?

 **Rosco McMahan** 21:33  
Yeah, that's that's roughly accurate, yeah.

 **Bhaskar Verma** 21:36  
Do we? Do we like once all the operations are done for the transaction, do we send in a call back then or for each operation do we send it back?

 **Rosco McMahan** 21:46  
This is this is probably a ludes me. I believe it's every batch I could be wrong though.  
So a batch of 100 gets sent.  
I believe could be wrong.  
Alright. Do you know offhand, Carrie?

 **Carrie Zhuang** 22:06  
Sorry, what?

 **Rosco McMahan** 22:07  
And thanks.  
Do you know if do you know when we do a callback? Do you know if it's per transaction or per operation?

 **Carrie Zhuang** 22:16  
Transaction.

 **Rosco McMahan** 22:17  
OK, there you go. That was wrong.

 **Bhaskar Verma** 22:22  
And this callback URL that you called back to. Is it something that we sent to you because I'm just not aware?

 **Rosco McMahan** 22:28  
Yes.  
Yes, yes.

 **Bhaskar Verma** 22:30  
OK.

 **Rosco McMahan** 22:31  
So the Me pub ESRP File Manager code creates the callback based on where it's currently what cluster it's currently living in.

 **Bhaskar Verma** 22:31  
OK.

 **Rosco McMahan** 22:41  
And then it it sends it off. Now if it's, if you see that you're not getting a call back, it might be because you put null there and I don't think we throw an exception or anything if it's null, we just allow it to do that simply for test purposes.

 **Bhaskar Verma** 22:56  
OK.  
OK.  
Yeah, I think those are that was really helpful just to understand the basics of it and.

 **Rosco McMahan** 23:03  
Yeah.  
Yeah. Yeah. So you mentioned your your stuff is not being updated.

 **Bhaskar Verma** 23:07  
Yep.

 **Rosco McMahan** 23:11  
Umm yeah. Do you know if that's because you didn't supply a callback?

 **Bhaskar Verma** 23:12  
Yeah.  
So actually I was checking today on the day I was working in the. OK, let's, let's let me first stop the recording and then we could discuss.

 **Rosco McMahan** 23:23  
Yeah.  
Sure.

 **Bhaskar Verma** stopped transcription