**Part 4 [Deprecate reddog, ISO]**

July 31, 2023, 3:36PM

57m 1s

 **Jacob Kissel** 0:03  
But yeah, So what?

 **Mohit Gupta** started transcription

 **Jacob Kissel** 0:06  
I'm going.  
What I'm presenting is specifically moving away from Red Dog for autopilot and Pilot Fish Publishing.  
Umm, last time we we talked about how we published into a PDF and it was through Red Dog and I'm I started this with a kind of an overview of the data flow just as a refresher as well to to illustrate specifically what's changing UM about the flow.

 **Dina Helal** joined the meeting

 **Jacob Kissel** 0:33  
So our responsibility as me pub is to get the images from SMF into the deployment teams environment in AP and PF and also as a reminder, data deployment is a system owned by a team.  
That's kind of a generic app team and the deployment team themself is a separate thing, which is an HCC team who kind of owns ask native and APPF deployment for Windows specific vhds out to their respective environments.  
So there's two different things.  
One is deploying the images out to the other APPF environment teams and pest native teams, and then one is data deployment which is a generic flow of data from outside of APPF into the boundaries of APPF.  
So uh, in this flow we start with SMF.  
They create the VHD, the compressed VHD's, which are known as whims.  
They're file extension is whim, but is really just a compressed VHD that the deployment team inside of APPF decompresses it and then pushes it out.  
Compressing it is important because that way it's a lot smaller footprint during the file copy through the data deployment system, so they give us these files using artifact service drops as me pub.  
We then push the files into the data deployment system and we kind of rely on the data deployment systems contract to actually get the files from inside of there into the deployment teams environments on autopilot and pilot fish machines.

 **Shailja Joshi** joined the meeting

 **Jacob Kissel** 2:10  
So skipping the data deployment system for a second just to finish the flow once the deployment team kind of has ownership over these files on their machines, they do flighting and deployment out to their rest of the bunch of machines in autopilot and pilot fish that use windows.

 **Dhiraj Singh** joined the meeting

 **Jacob Kissel** 2:26  
So they are actually kind of controlling the rollout of the Windows vhds that we give them.  
So they have a little bit of work which is decompressing them and then the rest of their work is focused on flighting them out to the rest of the machines in their respective environments for autopilot and pilot fish.

 **Eduardo Toraya** joined the meeting

 **Jacob Kissel** 2:46  
So basically, if we were to do the the most high level possible understanding is we need to find some way to make our files available on the machines of the deployment team in Autopilot and pilot fish and the the next level down is that we're going to use the data deployment system to do that and then the final, most detailed level without getting into like lines of code is how we're using the data deployment system.  
And that's in this middle circle.  
There's two methods, the original or.  
There's more than two methods.  
There's many methods actually, but the original method we have been using is the Red Dog method and the method we are going to be moving to is the Stratus method and these are two implementations that the data deployment system themselves make available to its customers.  
So there's tons of teams that publish data into autopilot and pilot fish.  
I think like Xbox is one of the bigger teams that has machines in these environments, so you can imagine the different type of data that they have to get onto their machines.  
There's tons of different ones, but so for us the original method was Red Dog, which is what we went over last time where we copy files onto the reddog file share which is on corpnet.  
So we need express route network one ES agents to perform that file copy and then we put an AP signature file there which is a manifest stating hey, these are the context contents of the stuff we wanna put into autopilot and pilot fish.

 **Nabajit Pathak** joined the meeting

 **Jacob Kissel** 4:19  
And that manifest is the point of contact for the data deployment system.  
To see that we actually want to do a deployment.  
So we can put any files on Red Dog and nothing will happen.  
But as soon as we put this specific AP signature manifest file onto REDDOG data deployment system will pick it up and see that we want to perform and deployment.  
It will read that file in any of the files inside of that manifest.  
It will take those and put them into their respective environments where it needs to go.  
So that was the data deployment system.  
One of the methods they set up to enable teams to get copy files into the environments.  
The next method is Stratus, which is kind of its own small team inside of the data deployment.  
Kind of radius of implementations that they enable and Stratus is much more hands off on our on our side, whereas we just have to call a Stratus pipeline extension with an ADO drop or sorry, an artifact services drop or VSTS drop and we kind of point it to that.  
And then after we give it that drop, the Stratus extension or the Stratus services themselves actually will take over the copying and do everything that's needed to copy those into the autopilot and pilot fish environments.

 **Alexey Loginov** joined the meeting

 **Jacob Kissel** 5:42  
Umm, so on the next page it's just a little bit extra info about the differences here.  
With Red Dog, we have to do 1 file copy onto reddog.  
We also have to do we have to create and sign the AP signature file and place that onto reddog and after the second part is done, the data deployment system is able to do the file copy into autopilot and pilot fish for Stratus.  
On the other hand, uh, we take all of the different files that SMF gives us.  
We put them together into one single artifact service drop, and then we call the Stratus ADO Pipeline giving it the name of that drop and also the name of the environment we're trying to deploy into.  
And then the Stratus, uh services will do everything for us.  
Umm this side I I couldn't find a much more user like readable way to illustrate all of these different ideas or the different options here.  
But what we're looking at on this slide is the deployment DOT I and I file and this is kind of defining where a given application inside of autopilot and pilot fish is getting chunks of data from.  
So basically the data deployment system reads a teams deployment dot ini file in that instructs it on how to get different pieces of data and where to put them.  
OO N the left side of one of these, let's see if I can draw on this.  
On the left side of one of these is the name of the deployment rule, and later on in a teams application, they'll say hey for this deployment rule put into this folder, but those those rules all look the same.  
So all we really need to know is that we define a deployment rule and then we define where it gets its data from and then after that is stuff that's defined by the data deployment team that says hey for deployment rule that you're getting from source depot, it needs to look like this.  
Point us to where and source depot.  
You're getting it from all the folder extensions and everything.  
So for Autopilot SQL 2019, they're actually doing a data deployment through source depot instead of through Red Dog or Stratus or something like that.  
This is an example of one of our own data deployment rules and it's using reddog, so this deployment rule says there's an SMB file share, so that's kind of the preface saying hey, data deployment system, this is what we're working with and that file shares location is this red dog.  
And then all of the child folders, and then there's an extra variable here which is signature as manifest.  
Basically what this is saying is we're signing our AP signature file with ESRP.  
There's another way to do Red Dog data deployment, where you manually sign it with like a user, like a developer's cert, smart card cert, and that would have a different extension.  
Here I saw Antonio raised his hand.  
Was that a misclick or I?  
I don't know if it's still up.  
I can't see very well.

 **Antonio McMichael** 8:49  
Are you?  
No, it was intentional.

 **Jacob Kissel** 8:51  
OK, sweet.  
What's up?

 **Antonio McMichael** 8:53  
Umm so I wanted to know who actually provides us with these rules.  
Is it something that we maintain and partners counts us and say, hey, we need to ensure that we're putting this content in this location.  
Uh, like as, as if we were handling all of the targeting?

 **Jacob Kissel** 9:07  
Umm.

 **Antonio McMichael** 9:10  
Or is this some type of share configuration file where there are multiple teams that can come and make changes and it's like maintained and like some type of git repository or something like that?  
Just trying to understand who can change the file and how do we ensure the accuracy of the file among all the partners.

 **Jacob Kissel** 9:21  
Yeah.  
For sure.  
Yep.  
So the so basically the how should I say like the setup of this file deployment dot ini file is defined by the data deployment team so they say if the data deployment file has an SD rule it's gonna be read like this and the data deployment system is gonna read like this.  
So the overall contract that you have to follow is kind of they set it forth the next level down is any number of teams can have their own deployment DOT I and I file and those are all stored in a source depot actually.  
So the autopilot and pilot fish environments are very old, I think.  
I think they actually predate Azure, so they their configuration files are stored in a source depot and anytime you want to make changes to it, you have to commit a change into a source depot repository.  
So our partner team, the deployment team from HTC, they have their own environment, their deployment dot I file is stored in a source depot and to commit to that source depot you have to also sign the commit with your own developer smart card.  
So it's even though it's on this old system, it's still very secure.  
I couldn't go in there and sign it myself because me as a user my developer certificate that I would try to sign it with wouldn't be allowed to because it's a locked down like that.  
So for the next question, who can make changes to it?  
It's defined like that.  
I have to basically if we wanna make a change to it, to add a new deployment rule, I would go to one of the teammates name to Milind.  
He's normally who I work with on the deployment team, but there's a couple other people that can make changes as well.  
I would say, hey, this is the rule that we want to change or add or whatever and he would commit to this deployment DOT I and I file the new rule.  
So it's a a contract that's set forth by the data deployment team on how you write the rules.  
And then underneath that, there is another contract between us as me pub and the deployment team as the application inside of APPF that we need to figure out which rules to put forward so that our files can get into their system.  
So previously basically we would say, hey, we're gonna try to use the Red Dog method to deploy some RS5 APPF images into your environment.  
So given the documentation, it says this is what the deployment role should look like.  
It needs the SMB and then it needs like the location in Red Dog and then it needs the signature as manifest tag at the end.  
So we would go to them and say, hey, this is the data deployment rule we need so that we can put the files and then they would basically name the data deployment rule something like SMF RS5, underscore image and then later on in their configuration they just say SMF RS5 image data deployments go to this folder.  
Umm, so there's a couple of different ones here.  
And then the new rule here is at the bottom for Stratus that before I go to that one, did that kind of answer the question about the ownership and like change logs of this file.

 **Antonio McMichael** 12:42  
Yeah, you actually beat me to the the real question I was gonna ask is how do we control the rollout when we modify our strategy and change methods from Red Dog to Stratus.  
But it looks like you're just going straight into it without even having to ask.

 **Jacob Kissel** 12:55  
Oh, yeah. Yeah.  
So yeah, exactly.  
We basically just have to.  
It's a handshake between US and the deployment team, and since we're the ones that are owning, getting the files over there, we're kind of the owners of the knowledge on how that happens.  
But they're the owners of the config file in their own environment, so we ask them to make this change basically.  
So the change here is we'll be right now, it's still in the final proof of concept phase.  
So we just have a test rule, but once we finish all of our code will be migrating these existing rules and we'll just point them to a Stratus endpoint instead of a red dog endpoint and everything else doesn't need to be changed.  
It's just seamless because the end result is the file still end up on their folders and the only difference is the data deployment implementation is just doing it differently.  
So for Stratus, uh, the rule looks a little bit less human readable, but basically VSORM and then all of this stuff up through here is always on a stratus rule.  
Easy deployer.  
API DOT traffic manager that's kind of the API endpoint for Stratus itself and that's telling.  
The data deployment system to read information from that and then the extension on this is the name of the environment we're going into.  
If we were going to into pilotfish this section right here would say PF gold instead of AP Gold.  
It's just one of the implementation details and then the very end is the name of the Stratus folder that we want to deploy and that'll be shown on the next page.  
Umm.  
In the actual Stratus implementation.  
So this is the new deployment rule that we'll be using and then I yeah, this is the last page for Stratus.  
I just wanted to show what the pipeline looks like.  
Basically, there's a build pipeline which is up here on the left and then a release pipeline which calls Stratus as it's only thing.  
So the build pipeline has a.  
It's goal is to take in any number of drops from SMF.  
Combine them into a single drop and then publish that.  
Drop.  
That's readable by Stratus and then later that drop gets actually pushed into Stratus.  
So it'll initialize the drop.  
For anyone unfamiliar with Artifact services, a drop is kind of like a bundle of data, and we're getting our drops from HCC from artifact services.  
But we also are using them here for Stratus as well.  
But The thing is like for example for RS5 there may be 5 different images that we wanna publish.  
Each one of those comes from its own drop, and for Stratus you can't give it five different drops and we also don't need some of the extra files in those drops.  
So for us, we are combining them all into one drop.  
The first job is initialize a new drop using a.  
All of this is using drop.exe, which is a tool that the artifact service team makes.  
Umm.  
And as I as I convert this into a more production ready pipeline, a lot of these tasks are gonna get squashed into a new task group that can do multiple things.  
So I don't have to keep reinstalling install drop deck you exceed and all of these jobs.  
But anyways, the first thing you have to do is initialize a, drop.  
The next thing is this job is actually a parallelized job that runs for every single drop that we pass into the pipeline, and it will download the drop from SMF.  
It will then push the files from that drop into the drop that we initialized earlier on in the pipeline.  
Umm.  
And it also deletes anything that's not a Wim file.  
So there's a couple, uh, like logs and digest files and things like that that are generic to HCC that they create for everything.  
And we don't need to give those to the autopilot teams.  
So in order to reduce our footprint of COP files copied into there, we delete anything that's not a Wim file.  
Umm, the next step is we also create the AP signature manifest.  
Still, this step is optional because it's not required for Stratus, but we're still in in the investigation of whether the deployment team is using that as well because that file does end up on their machines and there's a chance that their system put it dependency on it.  
So I'm still talking to Milan to figure out if that's required or not, but either way that's just another file we pushed into the drop.  
Lastly, we tell the drop.exe that we want to finalize the drop and then we create a drop metadata file which points to the name of the drop that we finalized and we publish that as a build a pipeline artifact.  
And then on this, on the release side for the data deployment task for Stratus, I'll go ahead, Antonio.

 **Antonio McMichael** 17:46  
I know.  
So I was gonna ask about the download step.  
So, like how big are these files?  
Are these like the 127 gig files?

 **Jacob Kissel** 17:53  
Umm, so for past native the files are are a lot bigger, but for APPF the files are generally not bigger than 30 or to 40 gigabytes.  
Umm.  
As far as I've, as I've seen individual and it's all parallelized, each one's running on its own agent, so it's pretty quick actually. Umm.

 **Venkatnagaraju Goursetti** joined the meeting

 **Antonio McMichael** 18:14  
Got it.  
Yep, that's how I was gonna ask.  
I know if we're going from copying to a file share that is using SMB to now have to download and then upload.  
I just wondered if that adds to the processing time, but if your sign is pretty quick that would be my only question about it.

 **Jacob Kissel** 18:26  
Mm-hmm.  
Yeah, it's generally pretty quick and the other benefit is that it's all parallelized now too.  
When we were copying the files onto the file share, I think it's actually slower because umm, it has to download.

 **Antonio McMichael** 18:44  
Bandwidth.

 **Jacob Kissel** 18:46  
Yeah, it has to download onto the agent and then do the file copy from the agent onto the file share when we had it downloading directly onto the file share, it actually the download would crash all the time because it was very long running.  
So we had to slow it down a little bit by downloading it onto the agent 1st and then copying from the agent onto the file share.

 **Antonio McMichael** 18:58  
Yep.

 **Jacob Kissel** 19:05  
So even that was a pretty long running.  
Umm, but yeah, that's that's the case here.  
The the files are normally not over 30 to 40 gigabytes of piece because they're compressed vhd's and it actually has it not been noticeably long at all.

 **Antonio McMichael** 19:21  
Got it.

 **Jacob Kissel** 19:21  
Umm.  
Compared to passing native pass native takes forever for their downloads.  
Umm.  
And then, yeah, so after everything is in one drop, we push that drop into the Stratus task on a release pipeline.  
Umm the Stratus Trigger connection is basically a service connection that we set up that has permissions to call the Stratus endpoint.  
It's all authenticated.  
We had to onboard to Stratus and they kind of gave this connection permissions to call them.  
The next one is the environment which that we're publishing into.  
So that was that either AP Gold or PF Gold Autopilot virtual environments and then data folder alias is the interesting one.  
That's the like, SMF RS5 or SMF W 2022.  
You all of these are variables that we pass into the release pipeline so that we can call the pipeline for anything server 2022 or RS5 and then Autopilot or pilot fish and then the data root folder and the folder filters are not necessary because we are doing that when we construct the drop itself.  
If we were in a more spread out system where we're getting drops from a different team, we would have in doing our own filtering, we would probably have to use these.  
But since we're taking the drops and putting them all together into one and filtering it on our own, we don't have to do any of the filtering on the actual Stratus task.  
But that's as simple as it gets.  
Basically, on the Stratus side, instead of doing file copies and making the AP signature file, we literally just give it the drop name, the environment name and then the data folder alias and it does everything for us, including signing.  
Because the Stratus task later on, I didn't add this slide.  
But basically you can define a stratus rule to only come from a certain pipeline.  
So even if another team tried to deploying to this environment using Stratus with their own service connection, Stratus would decline it because we say only pipeline with ID 1234 can deploy to this certain data rule.  
So when we when we go into the production set of these rules, there will be an extra piece at the end that says only pipeline 1234, which is whatever one we set up for production can deploy to it.  
So that's another benefit of Stratus as well.  
Umm.  
And then I think that's it for the autopilot Stratus related slides.  
I can keep sharing for kruti if you want me to click through or you can take over.  
It's up to you and then if anyone has any questions about APPF, go for it now as well.

 **Achal Shah** 22:00  
Yeah.  
Hi Jacob.  
So the the one on the left is something we on right?

 **Jacob Kissel** 22:04  
Yeah.

 **Achal Shah** 22:07  
And then that invokes what's on the right is that that how it works? Yeah.

 **Jacob Kissel** 22:12  
Yeah.  
So the left is a list of build pipeline tasks and then the right is an actual release pipeline task and we own the build and release pipeline.  
But on the right the Stratus task is a task that the Stratus team built and we pulled into our stuff.

 **Achal Shah** 22:23  
OK.

 **Jacob Kissel** 22:30  
But on the left, it's all our own custom things.

 **Achal Shah** 22:30  
Got you. OK.

 **Jacob Kissel** 22:33  
The reason why uh it's kind of split up like this is I think originally dated deployment was built, uh if if you guys are familiar with the.  
One yes, like managed pipelines.  
Entirely.  
Uh, the they have built in a lot more built in support for Stratus where they can even like they can build a set of data that's immediately gonna go through Stratus and it's kind of all managed and automatic.  
But for us, since we're not getting our built stuff out of A1ES build pipeline, we're getting it from another team that gives it to us with drops.  
We have to kind of do our own aggregation step, but still end up calling the same Stratus release.  
So we own the execution of all of these things.  
The build pipeline is more custom and then the release pipeline.  
We're just calling a Stratus task that was built.

 **Achal Shah** 23:30  
And the and the left hand side represents ingestion I.

 **Jacob Kissel** 23:35  
Yeah.

 **Achal Shah** 23:35  
Or is it OK?

 **Jacob Kissel** 23:35  
Yeah.  
So on the left hand side, if we finish this, it still doesn't get into Stratus.  
So all of that stuff can happen during ingestion.  
We can download and upload everything and have one single drop ready to go and then when publishes called we'll pass that drop into the release pipeline and we won't have to do any file copies on our side.  
All we'll have to do is wait for Stratus to do the file copy from our drop into the autopilot and pilot fish environments.  
Umm, I think there's one.  
There's two more hands up.  
I see Alexey has one.

 **Alexey Loginov** 24:16  
Yeah.  
Can I ask quickly like so the the pipeline we have is mostly to satisfy the Stratos contract, right?  
They expect one drop.  
We have more than one.  
We need to create a drop for them.  
This is what the left pipeline basically is doing right and and umm and I, I missed the beginning.

 **Jacob Kissel** 24:31  
Yep.  
Yep, exactly.

 **Alexey Loginov** 24:37  
What is the contract actually?  
So they provide a ADO task.  
Uh to call them.  
That's fine.  
How do we know they are done?

 **Jacob Kissel** 24:46  
Umm.

 **Alexey Loginov** 24:47  
Or if there is any failure, what is the contract here?

 **Jacob Kissel** 24:52  
Yeah, actually.  
So this is an improvement over Red Dog actually for Red Dog as soon as we push the AP signature file onto the Red Dog file share it, we don't really have another way of checking once the files are done it it's kind of just like we finished there and then the the team inside APPF the files will eventually get there.  
But for Stratus it's a little bit more advanced because this built in release task that they made for calling Stratus.  
It actually also has monitoring inside of it, so when we kick off this Stratus task, it won't succeed until the files actually are on the autopilot and pilot fish system.  
And that's something that the app teams have been asking for us for a while as well.  
It's just with Red Dog.  
It's a lot harder to accomplish something like that, but with Stratus it's all a really stratus is kind of like the ideal method that the data deployment team wants people to be using because we obviously everyone needs to move off of SMB file shares.  
They also want to move off of source depot for data deployment.  
That's even older than the SMB file shares for deployment reasons, so Stratus is a really well built out service.  
And yeah, this pipeline task will literally monitor the monitor, the deployment and it won't finish until the files are actually on the dependent teams machines.

 **Alexey Loginov** 26:14  
OK.  
So we call it.  
It's gonna be in progress until it's done, and we're gonna have some monitoring and play saying it's taking longer.

 **Jacob Kissel** 26:18  
Umm.

 **Alexey Loginov** 26:21  
Here is your ICM follow up.  
That's the strategy.

 **Jacob Kissel** 26:26  
Yeah.  
So we'll have the pipeline task itself won't.  
It'll just either time out or fail if if there's something wrong with it, but our service if we go one level back our service fabric services that are doing ingestion and publishing and stuff, they're calling this pipeline, that service will have a certain timeout where to expects, hey, normally Stratus is only taking an hour to deploy, but it's been 2 hours.

 **Alexey Loginov** 26:50  
Umm.

 **Jacob Kissel** 26:52  
So let's raise an ICM and then that ICM will point to the link for the pipeline that's doing Stratus and say, hey, this is the Stratus pipeline failed, go see if it has any logs for the DRI to check Oregon, things like that.  
And we'll be able to monitor the pipeline task itself as well to get early signals on if it's failing or taking longer than it should or stuff like that.

 **Alexey Loginov** 27:15  
OK.  
Thank you.

 **Achal Shah** 27:16  
So sorry.

 **Jacob Kissel** 27:17  
Yeah.

 **Achal Shah** 27:18  
Wait, quick follow up on that.  
They don't have any kind of like, do they always know from us like they don't have internally any alerting for problems?

 **Jacob Kissel** 27:31  
So that's a good question.  
I haven't looked into what their internal kind of like self ICM registering is.  
Umm, I will definitely take that as a follow up to investigate what kind of stuff they what kind of alerts and metrics they raise on themselves.  
Umm, they definitely should have something.  
So yeah, I'll definitely take that one.  
Uh, let Mohit.

 **Shailja Joshi** left the meeting

 **Mohit Gupta** 28:02  
Yes, I had a question around.  
So you said and this is what I understood and correct me if I'm wrong.  
So what I understood was we have to me pub has to copy from the artifact store of SMF and copy it into another artifact store which we give it to status.

 **Jacob Kissel** 28:20  
Umm yeah, that's correct.

 **Mohit Gupta** 28:22  
Is that correct?  
Is there a possibility to point Stratus to SMF artifact store?

 **Jacob Kissel** 28:33  
So the the sorry the the reason we can't do that is because the Stratus contract is they're expecting one single drop and for any given product, there can be multiple drops from SMF.

 **Mohit Gupta** 28:33  
Would that be that make things faster?

 **Jacob Kissel** 28:49  
So yeah, so for RS5 there could be like.  
You.  
Uh, I don't really know the names of the different drops they use, but there could be like 5 different products for RS5, but we do all of that in one and next month there could be 1/6 product for RS5.  
So if we wanted to just point them to the RS5 drops, we would have to have one data deployment rule per drop, and every time we add another product to RS5, we or another type of image to RS5, we would have to have an another data rule.  
But the the the system right now is that there's one data rule for RS5 and that way if there's five drops or six drops, all of them get aggregated into one drop.  
And Stratus can only take in one drop.  
So that's kind of the way it's set up right now.

 **Mohit Gupta** 29:36  
It is it it one drop per product.

 **Jacob Kissel** 29:41  
Yes.  
Yep, so RS5IN autopilot as one drop RS5IN pilotfish is a different drop server 2022 and autopilot is another drop.  
Server 2022 and PILOTFISH is another drop, so the product plus environment combination makes up one drop.

 **Mohit Gupta** 30:01  
Alright, that's someone.  
You mean the update type product plus update type or environment?

 **Jacob Kissel** 30:07  
Uh, I yeah, I guess environment and update type are synonymous because AP and PF are two different update types, so each one is tied to 1 Yep environment.

 **Mohit Gupta** 30:15  
Yeah.  
Got it.

 **Achal Shah** 30:23  
Hey, related related question in theory could SMF do this copied in the in in the one location that the deployment uh expects so that like they would do it as soon as the media is ready right?

 **Jacob Kissel** 30:24  
Umm yeah, go ahead.

 **Dhiraj Singh** left the meeting

 **Achal Shah** 30:44  
So the image is ready.

 **Jacob Kissel** 30:45  
Sorry you asking if we are asking if SMF could give us one drop with all of the products combined instead of all the different groups.

 **Achal Shah** 30:52  
Yeah.  
So that so that me pub doesn't have to do this is that, I mean is is there a reason why that's not done?

 **Jacob Kissel** 31:02  
Uh, so we did not ask that question.  
Umm, we can definitely ask them to see if that's something they can do.

 **Achal Shah** 31:05  
OK.

 **Jacob Kissel** 31:08  
Uh, I guess the assumption was just because historically it's always been one drop per thing.  
I don't know if it maybe that's an efficiency thing on their side or maybe it's more flexible for them because the different ones get finished at different times or get built by different systems.  
But I can definitely ask them for some more information on why they segregate or split up the drops that specific way rather than combining them.  
That if there's no other questions, we can go ahead and move, move forward.

 **Mohit Gupta** 31:56  
Sure, sure.

 **Jacob Kissel** 31:56  
Uh kruti did you?

 **Mohit Gupta** 31:59  
You're 13-8 minutes into it.  
I'll be planned about 10 minutes.

 **Jacob Kissel** 32:02  
Sorry, that took a lot longer than I expected.

 **Mohit Gupta** 32:04  
So I think that that that is a indication of how complex things are in media publishing.  
I'm sure other systems are also complex, but what I find is whenever we start discussing about something and it never finishes in 10 minutes, I don't know.  
It's common for all topics.  
It's not just this one, Jacob.

 **Jacob Kissel** 32:24  
Yeah.

 **Mohit Gupta** 32:25  
And anyway, So what we'll do now is.  
I will go over with ISO deep dive kruti.

 **Kruti Kallola Mohanta (WSD)** 32:37  
Yes, OK.

 **Mohit Gupta** 32:38  
Yes.  
OK, over to you Kriti then.

 **Kruti Kallola Mohanta (WSD)** 32:40  
OK, I'm selling my screen.

 **Mohit Gupta** 32:43  
Yeah.  
And and and just just a thing.  
Don't feel stressed to complete everything in the remaining 20 minutes we can we can spill over some of your content to next week if you want, so don't be pressed or finishing everything.

 **Kruti Kallola Mohanta (WSD)** 32:56  
Yes.  
Sugar.  
As many as skin visible.

 **Mohit Gupta** 33:28  
Yes, it is visible.  
I think you need to go to.  
Four or five slides ahead.

 **Kruti Kallola Mohanta (WSD)** 33:35  
Yes.  
And thanks Jacob.  
Now we are starting on the I I saw part.  
So the first step is that why I like, why do we have ISO monthly refresh as media type?  
So I saw is that the media type which needs to be published to below channels.  
So there are two parts to that ISO media type.  
One, we can say ISO monthly replace, which is a generated every month and it is published to VLSC channel and BS channel and we pub comes here like uh we pub responsibility is to make sure that ISO Media is published to this required channels and another category is before GA and Pre GA publishing and that is done for OEM and RTM partners.  
And this and media publishing doesn't play a role in this.  
So as means we give the source of truth.  
But again, SMPS media and by email verification like email, some email is sent and really manager generally take these data and it is published via other systems.

 **Dina Helal** left the meeting

 **Dina Helal** joined the meeting

 **Kruti Kallola Mohanta (WSD)** 34:48  
So far, from your point of view, we just focus on the ISO monthly refresh and our objective is to publish the ISO media updates to BLS channel and BS channel.

 **Tavishi Gupta** joined the meeting

 **Kruti Kallola Mohanta (WSD)** 34:59  
So this is can sort of track it where you can see that we have had some monthly refresh media type and these are these are the, these are the media that is required to be published.  
Now coming to what does me pub do for ISO publishing?  
So like for other update types here also we have the same five steps and create, ingest, publish, approve, go live and create a same with uh is exactly same uh what is done for other update types like PAND, APPF container that we discussed in the previous session.  
So we won't be discussing on this because it is exactly same and the call live step, it is no offs, there is nothing, there is a dummy step and there is no step involved from pubsuite.  
For ISO it is just a bypass and we just completed the logic here, not only in the.  
In the next slides, we'll discuss from the ingest, publish and approval steps and how it is little bit different from other update types.  
And before we talk about ISO or other types, let us little bit discuss about the software part numbers.  
So what is software part number?  
I've given an example here like X-23 some value, so this is a software part number are associated with the cost to Microsoft and so we cannot create multiple.

 **Venkatnagaraju Goursetti** left the meeting

 **Kruti Kallola Mohanta (WSD)** 36:22  
We cannot consume multiple partners as in when we like.  
Instead, we have to there is a process and like RSM which is owned by the PRSS or uh it is responsible to make sure that the there is a process where we get the part numbers and every time we consume a part number it is incurring a cost and this is a screenshot of release UI ESRP release and here is we can see for RSM religious there is a part steps for RSM Reserve part number and in this step the result this part numbers are blocked or they are the status is changed and once this step is there is a process for cancellation but once this step is done we cannot use the same part number for other release ID S or other religious or else in ESRP it will fail with the error that this software part number is already consumed and we cannot like resubmit another release with this.  
So here is the concept of partner main thing about is that this is associated with a post and these are limited in numbers and and we cannot just spin up multiple part numbers.  
Yeah, out of anything.

 **Yashasvini Rathore** left the meeting

 **Kruti Kallola Mohanta (WSD)** 37:31  
Instead, there is a process we need to follow to get these part numbers.

 **Yashasvini Rathore** joined the meeting

 **Kruti Kallola Mohanta (WSD)** 37:38  
Now coming to ingestion part, so in in a high level overview we can say that in ingestion 4 ISO there are three steps, fast is fetching data from SM.  
So this is same with other update type also like other update type also we consume the data from SMF.  
So in that payload we get like in the payload that we pop.

 **Yashasvini Rathore** 37:59  
Prateek, a solid under I think your screen is stuck on one presentation.

 **Achal Shah** left the meeting

 **Yashasvini Rathore** 38:05  
I tried rejoining even then it is on 8 slide.  
Is it the same for others as well?

 **Achal Shah** joined the meeting

 **Kruti Kallola Mohanta (WSD)** 38:11  
I'm currently on 11th side is other for others.  
Is it on 11 side in just in?

 **Achal Shah** 38:20  
Yeah.

 **Mohit Gupta** 38:20  
No, we're still on 8th.

 **Achal Shah** 38:20  
I just had to rejoin.

 **Kruti Kallola Mohanta (WSD)** 38:23  
Ohh OK, let me reset my screen then.  
OK.  
Is a.  
Is it visible now my screen?

 **Mohit Gupta** 39:08  
Yeah. Yes.

 **Kruti Kallola Mohanta (WSD)** 39:11  
Thanks Asia for letting me know. Umm.  
Yes.  
So I think that can sort of release you right that I was showing and here is the RSM Reserve part number step.  
Now coming to ingest the first step.  
Uh is same like we fetch data from SMF for that release proposal ID.  
So this is a payload that we get.  
So the first section contains the information about the SMF.  
So this is our payload identifier.  
This is the URL that we used to get all the data required for that release from SM and that there's a fast app and it is common to other update type as well.  
And the second part is fetching uh part numbers from AWE artifact service.  
So there is a second section with payload FCA where we have a payload and different.  
So this is the artifact ID and using this we face the media plan XML file which has the part numbers and the media images in media images name.  
So using that identifier like using this ID we get the we downloaded the file from the AW artifact service and we pass this part numbers and the last step is same as in other update types we update the same file this data in release proposal metadata document in Cosmos DB.  
So with that.  
Now coming to publish.  
So in publish.

 **Mohit Gupta** 40:56  
Sruthy we have a hand up.  
Do you want to take question now?  
Published do you want to go ahead with the question?

 **Tavishi Gupta** 41:16  
Uh, yeah.  
In the previous slide, Kruti do we get do you get that information from SMF or do you like it?  
Is that information coming from AWE?  
Uh for the for the the pillow it's saying.  
Are you getting it directly from SMF?

 **Kruti Kallola Mohanta (WSD)** 41:39  
Sorry guys.  
Uh.  
Disconnected in between, but I think the question was in the last slide.  
Uh, this affair fellow and pay for AWE article service are you getting it from AWE?  
SMF right.

 **Tavishi Gupta** 41:56  
Uh, so the payload identifier.  
Also, I think it's a, so I I think SMF creates it but it is from AWE right?  
Like the So media publishing services or one pub services talks to AWE and and then this gets from AWE.

 **Kruti Kallola Mohanta (WSD)** 42:11  
Correct.  
And it this is fetched from AWE, so AWE is the API where.

 **Tavishi Gupta** 42:17  
Yep.  
But even the payload identifier, right?  
Even the the the first part, there's an offer.

 **Kruti Kallola Mohanta (WSD)** 42:22  
Yes.  
Yes, this SMF part.  
This is sent during sign up.  
This parade and Paris sent and AWE passed that information to make up.

 **Tavishi Gupta** 42:37  
Right.  
Thank you.

 **Dinesh Vijayakumar** left the meeting

 **Kruti Kallola Mohanta (WSD)** 42:42  
Is many screen and visible.

 **Fabiola Ansara** 42:52  
Yes.

 **Tavishi Gupta** 42:53  
Sit is Yep.

 **Kruti Kallola Mohanta (WSD)** 42:54  
OK.  
Thanks for confirming.  
Yes, coming to publish for ISO.  
So in the publish step, uh in mizrak just for ISO we just submit to ESRP ESRP scan so that we submitted to ESRP scan via one pub ESRP and which internal submit to ESRP scan and we wait for ESRP scan to complete.  
So they say this is the pain point compared to other steps in ISO.  
So generally we see a lot of once we have been seeing issues in publishing for ISO and it is primarily because of this RSM part numbers.  
So what happens?  
Like I said in a previous steps, the part number cannot be like it has to be if we have.  
If we are submitting a religious part number, we cannot create another release.  
The same part number until analysis cancelled or done anything.  
But let us say in the pipeline, due to some reason due to any reason, if our uh well this is scan pipeline is broken or like there is any issue then in other cases generally what we do is get on the retry logic app and the publishing is steps are repeated which generally create a new release and the data is submitted means the same data is submitted with the release new release ID and generally there is no issue but in case of ISO again resubmit the same logic app it will create a new release again which will fail because the RSM partners are already send you.  
So I said, I said anytime any issue happen for ISO specifically, the DRI has to go to the cosmos DB or to other portals to make sure like they should know the domain knowledge and make sure that the whatever release ID has been created only there that is consumed.  
And the same thing I've used for like to successfully pass the release.  
So as in the screenshot, we can see we have a a like ESRP scan status and ESRP operation ID.  
So in case anything happens or any issue happens like we know that the religious past but due to some reason our logic apps stopped working or something happened then there has to manually go this and then update it.  
But the number of files as we can see are he was like sometimes 700.  
So that's a little pain point though.  
We have a script and DRI who has the proper domain knowledge.  
General, run the script to get it sorted, but this had been a little issue till now and a little pinpoint and something probably we need to work upon.  
But in summary, yes there are published.  
We just submit to ESRP scan and via onepubesrp and that's the publishing for ISO.

 **Mohit Gupta** 45:43  
Kruti we have a few end up, yes.

 **Kruti Kallola Mohanta (WSD)** 45:44  
Any question OK?  
Yeah, I see.  
If you can say sorry, what?

 **Eduardo Toraya** 45:49  
No, thank you.  
I I just wanted to ask Richie if you know if you knew more, more, more of something specific about what the errors we have been publishing for ISO at the moment, because I do know that we umm that we are deploy the fix is about to release as I go for this and I don't remember having seen a publishing issues for ISO.  
So if you could let me know if, uh one you know about, that'd be great.

 **Kruti Kallola Mohanta (WSD)** 46:11  
Yes, Sir.  
Thanks Sarah for bringing that up.  
I was looking recently, I, Sam and I see no recent ICM because of ISO.  
What I was talking about was like from last December till now, like in a six month time period.  
But coming to your point, yes, that recently the changes that has been made for moving to new ESRP that means instead of using the old ESRP like in the CR service to now ESRP we don't send any any.  
So in I like this publishing stamp for Fortress.  
So in fact in 6B, and I think the last week, there was no incident related to ISO at all, not all in producing anywhere.  
So I think with recent changes probably this has not occurred.  
But uh, yes, coming to like previous six month or something, we have seen few issues and I have given in the upcoming slide to sample ICM's.  
Yeah.  
Uh, where?  
Like I saw impacted because of publishing.

 **Eduardo Toraya** 47:10  
Thank you.

 **Kruti Kallola Mohanta (WSD)** 47:12  
Have you seen?

 **Tavishi Gupta** 47:12  
Uh, yeah, just like, so actually I kind of like, wanted it.  
Maybe this is already answered, but so at this point we haven't actually submitted to ESRP or have we submitted to ESRP.

 **Kruti Kallola Mohanta (WSD)** 47:27  
In the publish step, we submit it to ESRP actually ESRP release.

 **Tavishi Gupta** 47:33  
Then.  
Why?  
Why so it there is like in the new ESRP if I I think maybe we you guys are using it now, but like there there is a cancel step right?

 **Kruti Kallola Mohanta (WSD)** 47:49  
And right.

 **Tavishi Gupta** 47:49  
So, so so instead of like cleaning up everything we should, I think we should be using cancellation of ESRP submission before retrying this and then we can easily retry.  
Is that what is happening, Eduardo?

 **Fabiola Ansara** 48:05  
Yeah, I can talk a little bit about this.

 **Kruti Kallola Mohanta (WSD)** 48:05  
Yeah.

 **Fabiola Ansara** 48:07  
So yeah, when we introduce the part of the new feature, well, the new user P there's two new tasks.  
This expire all audiences and there's another one that calls inside the expiry all audiences.  
I call the cancel logic app, so actually we we exercise this in 6B.  
The cancellation was successfully done, so we have, I think my my memory is very bad, but I think was Windows 11 and it's like somebody 22H2.  
So I'm not very good with memory.  
Uh, but they, they retry.  
So we hadn't known issue in the cancellation because it was updating the incorrect like incorrect release proposal metadata document.  
I don't release document.  
I think it was updating the incorrect document but it rather was taking care of it.  
I and I think Edwardo can talk about more about this, but yeah, we're using, we like they can now resubmit the suburban numbers.  
The cancellation goes to the ESRP portal and we can see the status as cancelled in the portal.  
And yeah, the publishing, it goes like correctly.  
So we don't have to do any cleanup and we don't have to do any manual work anymore and go ahead with what if you have more details.

 **Eduardo Toraya** 49:34  
Thank you, baby.  
So I think Tavishi does have a different point and I don't think we support what she's talking about.  
So what happens now with the cancellation changes is that let's say we do have a a build error from HCC and a new churn has to come in.  
Previously we did not support it because it's parallel audiences did not work in automation.  
But now let's say we have we have the first turn, which does not require cancellation.  
That one goes through, but we need to we get another sign off and we need to to publish again.  
Then cancellation will work correctly, but let's say we have a publishing case where there's something like malware detected.  
Uh.  
Then, if there's no new turn and we need to resubmit, I think what's issues point was getting to was so that we could like cancel in the publishing step.  
Am I correct their sabishii like cancel before publishing in any scenario?

 **Tavishi Gupta** 50:22  
Right.

 **Eduardo Toraya** 50:24  
Uh, so know that, be that.  
Be a good improvement, but we don't support it at the moment.  
We do support cancellation when a new churn happens, so that will be on a on a full whenever there's a new sign off.  
And what what?  
What happened in the past releases as the in specifically that case, but in case that I I believe the only case I can think about where publishing does not work is if we do have a malware case and for that case we do.  
We would need to to do manual manual, so we would need to change the flow so that whenever we publish we cancel 1st and we don't have to go through the AWE AWE play steps.  
But that's not how it works at the moment.  
So we would only cancel with a new sign off.

 **Kruti Kallola Mohanta (WSD)** 51:11  
And cardim.  
Anything else in this else we can move to next slide.

 **Achal Shah** 51:19  
I had a quick question how the software part numbers?  
How do they get into our system?

 **Kruti Kallola Mohanta (WSD)** 51:26  
So it happens in the English step for the part, numbers are fetched from the media plan XML fight.  
But if your question is ready to like, why do they come to the media plan XML file?  
So I think that is and everything.

 **Achal Shah** 51:40  
No.

 **Tavishi Gupta** 51:41  
I I I can take I I could take that if that's OK. Yeah.

 **Kruti Kallola Mohanta (WSD)** 51:43  
Yeah, that's your.

 **Tavishi Gupta** 51:44  
So yeah, so with every release, we actually have a bootstrap like.  
So when we create release tickets, we have a bootstrap.  
We create a bootstrap play so it has a bootstrap activity which actually sends which is basically 35 days in advance.  
We actually send this email to POP people pop team sitting there, asking them to actually create the.  
Have the software part numbers ready associated with that release, and then when SMF signs off we have a ingest activity which basically makes a SQL call to and then uh to their database.  
The Home Depot database and I think you and hold you were actually kind of looking into it to actually make it make it part of PME.  
But they they it's it's not PM they they are not moving it to PME right now and then so we we we we call like we make a SQL a call get all of the part numbers and we store it and then with with and then in the same ingest step after we get the store we create an artifact umm in CAI in content agnostic ingestion that pipeline we create a media plan we associated that we associate these new part numbers with the with the media with each media in the media plan we upload it so that one pub can consume it and download it and in the next stage which is the the I I just before after the ingest.  
Yeah.  
Mohit, you have your hand up.

 **Mohit Gupta** 53:32  
Yeah.  
Is this handling of media Plan XML and uploading?  
Is this manual or is this automated?

 **Tavishi Gupta** 53:38  
Call or to be automated.

 **Mohit Gupta** 53:41  
Automated.  
OK, so fetching fetching the the software part numbers through a sequel, updating media plan and uploading it in the right place, all is all is automated.

 **Tavishi Gupta** 53:54  
That's correct.

 **Mohit Gupta** 53:56  
Yeah, that's change because we are building a media plan.  
Umm, no editing tool and we are updating the software part numbers manually.  
We're building a feature to Umm update the software part numbers manually so.

 **Tavishi Gupta** 54:13  
So what is not and I think this is what the difference is.  
I think what you guys are like what to do week is like the first template.  
So.  
So what we don't have yet is the very first media XML.  
So we start using this after RTM.

 **Mohit Gupta** 54:32  
Ah, I see. OK.

 **Tavishi Gupta** 54:33  
So you guys are like working on like just like art RTM level.  
So because that point, like it atoms have to create and give us like AWE and one pound of this media publishing team, or a template, a media plan template that is the first very first template we are going to use and afterwards we're going to keep on updating it.

 **Mohit Gupta** 54:38  
Got it.

 **Tavishi Gupta** 54:56  
So that is automated after that.

 **Mohit Gupta** 54:57  
But.

 **Tavishi Gupta** 54:58  
But like that very first media XML is not that that's a template and I think that's what you guys are working on.

 **Mohit Gupta** 55:06  
Yeah, got it. Thanks.  
And and we are one minute over. Kruti.  
Is this a good time to stop or do you want to stop at a logical closure?

 **Kruti Kallola Mohanta (WSD)** 55:19  
Umm one question.  
Ah, you have to team wins.  
We can end yet or we can approve.

 **Mohit Gupta** 55:28  
So what we gonna you can we can.  
What we can do is we can leave the remaining part of ISO and VHD for next time, which will be probably the last session where we go over the update types and after that we will jump right into the incident classes.

 **Sam Son** left the meeting

 **Kruti Kallola Mohanta (WSD)** 55:47  
OK, see you.

 **Mohit Gupta** 55:50  
Makes sense?  
OK, so we are two minutes over.  
Anybody has any questions?

 **Antonio McMichael** left the meeting

 **Achal Shah** 56:04  
He Mohit at some point will be also be doing like the software architecture while architecture of the system that kind of thing.

 **Mohit Gupta** 56:14  
Umm yes, we should do that.  
Thanks for reminding.  
I think we should do that.

 **Achal Shah** 56:22  
Yeah.

 **Mohit Gupta** 56:22  
Yeah.  
OK I I am.  
Fabio will work together on that.

 **Achal Shah** 56:26  
OK.

 **Mohit Gupta** 56:28  
Thanks.  
OK.

 **Kruti Kallola Mohanta (WSD)** 56:34  
Thank you.

 **Mohit Gupta** 56:34  
If more questions.

 **Kruti Kallola Mohanta (WSD)** 56:34  
Thank you everyone.

 **Mohit Gupta** 56:36  
Thank you all for joining.

 **Leopoldo Estrada Vargas** 56:36  
You know, everything was mine.

 **Mohit Gupta** 56:38  
Thank you all for joining.  
This is really valuable and hopefully we'll finish all the update types next time.

 **Leopoldo Estrada Vargas** 56:45  
Thanks. Wait.

 **Kruti Kallola Mohanta (WSD)** 56:47  
Thank you.

 **Dina Helal** 56:47  
Thank you.

 **Achal Shah** left the meeting

 **Mohit Gupta** 56:48  
Thank you all, bye.

 **Fabiola Ansara** 56:50  
Thank you.

 **Dina Helal** left the meeting

 **Nabajit Pathak** left the meeting

 **Leopoldo Estrada Vargas** left the meeting

 **Yashasvini Rathore** left the meeting

 **Jacob Kissel** left the meeting

 **Tavishi Gupta** left the meeting

 **Kruti Kallola Mohanta (WSD)** left the meeting

 **Alexey Loginov** left the meeting

 **Fabiola Ansara** left the meeting

 **Mohit Gupta** left the meeting

 **Mohit Gupta** stopped transcription