**UPS KT from Dublin team part 5 (ESD)**

October 10, 2022, 10:33AM

52m 16s

started transcription

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** joined the meeting

 **Lacastan Moodley** joined the meeting

 **Pinar Sen** joined the meeting

 **Dina Helal** joined the meeting

 **Kruti Kallola Mohanta** joined the meeting

 **Bhaskar Verma** joined the meeting

 **Dina Helal** joined the meeting

 **Mustafa Mizrak** joined the meeting

 **Kush Mishra** joined the meeting

 **Halyna Hladkivska** joined the meeting

 **Jelena Maksimenko (CPL Solutions)** joined the meeting

 **AB# Mishra** joined the meeting

 **Dina Helal** 0:11  
OK. So today we will go over the end to end flow of publishing ESC upgrades through movie 6 to WSS customers.

 **Kruti Kallola Mohanta** left the meeting

 **Dina Helal** 0:26  
I won't type too much into the AWE part, but I will show it as part of the flow and head in our Mustafa. Anyone wants to to add something, please feel free to do so.  
So let me share my screen. I'll go through some slides that we had a while ago when we started to work on this project.  
And some of them might be a bit too detailed, so we can skip over them or dive into them, depending on on what you guys want. And then I can show you the the actual system like we can we can go through the cycle together. So let me share my screen.  
OK.  
Umm yeah. So previously yes, these were published to enterprise customers, so to to WSD US through an an old system called when PR, which we appreciated when we start when we worked on this project. So basically we moved it to the to the new modern system which uses.  
A cloud based technologies. So everything moved to A to ASIC and our services are running in service fabric and.  
Uh.  
We are triggered by AWE using ipublish V4 API.  
If you want to know more details about any of those things, or if something you're not familiar with, please stop me.  
So in a single yes, the release ticket. So we publish, yes, the upgrades for Windows 10 and Windows 11 for Windows 10 we publish for A2 Editions, consumer and business, and we support those architectures. So for Windows 10, in single release ticket we publish 190 bundles, 190 updates with different.  
And targeting information based on which edition this update supporting which architecture and which language. So those 38 are 38 languages, so 38 languages for X64 for consumer, 38 languages for X86 consumer and so on.  
The difference between Windows 10 and Windows 11 is that for Windows 11 we don't support X86, so that's why we have only 114 bundles and what we've published to WSUS customers through the movie 6 NextGen pipeline is the.  
And the artsy I'm release. So the the first time we ever published this product, this is what goes through this pipeline. Then there is meeting refreshes, which are published to the cat. So those don't go through our system. What goes through our system is they're very first time we publish a product. So this happens like every three months or every couple of months. This is not agular publishing that happens.  
Every be week or or every month in January. This is something that that happens whenever we have a new Windows product or when we are refreshing this product. So if if something is if an issue is found and we need to republish it so then we we can publish it again to WSUS customers through movie 6 and the new bundles will supersede the older ones. So whoever got the old one will will be able to get.  
Then you one on top of it basically.  
Umm.  
And overall, so this is like the end to end.  
Flow.  
So it starts from the Windows builds.  
And which then goes through packaging and then go through the media creation process. So once the the media is is published.  
Through artifact service and the artifacts are in specific like on Prem locations. This is when our system starts to publish so we get the media from this location where it was published and we.  
We add like so we create the bundles which I will I will go through in in the next slides, but we we create the updates basically with the appropriate metadata. So those 114 or 190 bundles based on the product and then we publish them to pop suite and then.  
And then two WSD US, we used to support publishing tool, which is now not done through movie 6 anymore. So we only publish to to WSD right now.  
Do you have any questions on on this flow?

 **Halyna Hladkivska** 5:50  
Umm, actually I have a question. If the everybody is familiar with what what we ESD is what the cat is moon. So if everybody is familiar with the acronyms and terminology.

 **Dina Helal** 6:03  
That's a good point. Thanks, Selena.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 6:05  
No, I'm not. I was about to ask, what is ESD ESD is the file format.

 **Bhaskar Verma** 6:10  
But it is, you know.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 6:10  
Yeah.  
It's not just you.

 **Bhaskar Verma** 6:11  
I shall very kissable.

 **AB# Mishra** 6:20  
There's too much noise from the room conference room.  
It's much better.

 **Bhaskar Verma** 6:26  
I think he is. He is in older version of installing or particular operating system and a on a piece of hardware, right? Or as a use in IS or not wrong.

 **Dina Helal** 6:38  
Yes. So yes, these are like Windows upgrades, so.  
Well, and users can install, yes, these two upgrade from a for example Windows 10 to Windows 11 or from a specific version of Windows 10 to to a higher version.  
Uh, so yes, there is a form of media that is used for Windows upgrades.  
And.

 **Bhaskar Verma** 7:05  
And you mentioned that we publish it to WSD and we.  
Ohh I I I'm publishing the book so do we publish it as part of WS catalog?  
But.

 **Dina Helal** 7:19  
And so in in through the movie 6 pipeline, we published to W Sauce only. We don't publish to catalog but it gets published to catalog through another pipeline. As far as I know, I can get more info on that.  
Previously, for movie six, we used to support publishing tool as well, but now this moved to to the UPS system, so it it still gets published but not through the movie such system.

 **Bhaskar Verma** 7:49  
OK.

 **Dina Helal** 7:52  
And.  
Any other questions about the acronyms or something is not clear.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 8:02  
So how is?  
Well, what is special about ESDN?  
Are we not doing something similar for other media types?

 **Dina Helal** 8:16  
So we we are publishing so so there are different types of medias that get published depending on where what the audience is. So as far as I know we have like we published media to Azure marketplace and two different kinds of audiences and our enterprise customers get their upgrades through the W SUS server.  
And I'm not sure how they get other kinds of medias, or if this is the the only kind that day that get in order to support Windows upgrades.  
Sorry, I'm not sure if this answers your question.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 8:59  
OK, it does, yeah, my question is.  
Yeah, you said, right. Any time a product page just hard TM use this to do this, right? So it also includes the feature pack, right? That comes every six months externally, 22 next week.

 **Dina Helal** 9:07  
Umm.  
Yes, yes, yes, yes, yes, yes, yeah, I'll, I'll, I'll show you an example of when we published a Sun Valley tool, for example, and and so these days we have a. So we had Aga list that happened for SV2 on the 20th of September. So this is the A SV2G8 date we we published, yes, these two W users for SV2 and we have one that is coming soon for Windows 1022H2.  
Uh, which is on the 20th of October? I think so. This is being the media is being generated, so it's already generated and published for validation to dogfood more and will be staged to be published to live more soon.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 10:08  
So all of these are affected by release tickets. Again, right, talk about it.

 **Dina Helal** 10:11  
Yes, actually I think it will be better if I show you the the actual flow like I I can share the slides afterwards, they they might have too many confusing details.  
So for example, this is when we published Windows 1122H2.  
Last month.  
The way it starts so we I'm not sure how familiar you are with realistic. It's on AWE, but we do have a some level of like we we have automation for for the regular monthly releases on AWE. However for ESD's because those are not that frequent.  
We don't have this process fully automated yet, but we do want to to reach that at some point. So currently the way it happens is it starts by the creation of this CPT custom publishing task. So this gets created by.  
An arm and it includes the the build details. So after the media generation is done.  
This information is shared with us through this CPT.  
And it's passed that has like, which audience this is going to go to, what is, what is the TTGL date? What is the the product?  
And the next step is the creation of the release ticket on AWE, which currently doesn't happen by automation but happens manually. So someone on the AWE team goes and creates this release ticket. So this is the release ticket that corresponds to this CPT.  
And then if we go to the release ticket.  
Let me show you the details here.  
So we have something called branch type which can be current branch or active branch. Active Branch is appreciate. So we we publish for for a specific product twice we publish.  
Uh, and the preview release, which happens before it is released to.  
So are you tail to customers and this is the the the internal preview release, so the branch type in this case would be active branch and then then next release which happens a couple of months later is the the current branch, this is the one that that goes to.  
To everyone else.  
Umm for example.  
And if you're getting, this comes like weekly.  
Announcements you can see, for example, for Windows 10, this was the the preview date when we published Windows 1022H2 and this is the upcoming G8 date and for Windows 11 this was the the G date and then like one of the older emails we can see that this was the preview date. So the difference between the preview release and the GA one is the branch type.  
Active branch or current branch and what this translates to on on our side is in in the targeting information that we add on the updates. So we have something called detectoids that are associated to the updates and they have some logic that specifies what kind of machine can get this update and this logic changes based on the branch type to make it available to.  
Users on the Insider preview program or to be available to everyone else basically.  
Umm.  
And the information that we see here on AWE is like similar to what we have on the CPC, which got added to the release ticket. I can show you like if we are adding like how this creation gets done. So this is the AWE PPE portal.  
Because I can't create something on the prod ones. So let me show you if we start a new play.  
We need.  
To provide those two strings and those two strings are specifying the the the build string where we are getting the ESD from.  
Which is mentioned here basically.  
So those two locations need to be provided and then the play gets created and then they release ticket gets created in AWE. This is the currently manual part on the process. Once this is done then then everything else proceeds with automation.  
Any questions so far?  
No. OK. OK, so So what happens next is that AWE?  
Calls one pub.  
First it calls one pop 2 to ingest the media, basically to to get the media.  
And from artifact service and get all the build details and store it in our database and then.  
Publish it to test audience to publish it to internal audience.  
And then we have a themes who work on validating those so. So we publish those and we share the updates IDs.  
With the validation team and they have their own and this gets published to to dogfood more, so it doesn't get published to to live more gets published to the the Dogfood Microsoft update server. And those teams are able to connect to it and install those updates and make sure that the updates are applicable as they should be to the right like whoever.  
Supposed to be able to upgrade to. This is able to upgrade to it and whoever is not supposed to get this is not seeing it as being offered.  
Once this validation is done.  
And then there is this sign off that that gets triggered and makes the play proceed to stage or to publish the updates to to retail, meaning publish the updates to go to live mode.  
And and.  
Then those updates are created and published to to the life more server.  
But they don't appear for anyone to to be able to pull them or update to our install them until approval is also triggered. So we have to publish to test audience, get sign off, then publish to retail, get sign off and then.  
The last thing is that it needs to hit ttgl, so if approvals happens on the 19th for example, it's still not available until the 20th at 10:00 AM.  
So this is like the the full process of or like the the cycle that they release ticket goes through.  
And what this means on on the one pub side is that we have this.  
Something called the Elise proposal. So this release proposal is like.  
Umm, I got presentation of the release ticket.  
And and this release ticket on the one pub side has those 3.  
Operations that got triggered from AWE publishing to test audience publishing to retail and the the final approval publishing to test audience inside it has auto approve because this is publishing to internal teams for validation. So it gets published and approved automatically so they can.  
And pulled update and installed it.  
So if we take a look at at this one.  
What it does is that it creates only the updates of the English language, so you know we have like those. So this is Windows 11. We have 114 updates and those are all the different combinations of addition architecture and language. But when we publish for validation, we only publish the English language and those are the updates IDs that we share with the validation team. And once everything is done.  
Then we go and publish the full.  
The full list of all the different bundles.  
And and they remain in like scheduled state until they get approved right before the time to go live.  
All of this is visible on track it as well, so on track it. This is the system that managers on Gliese managers use to.  
To see overall, like all the ongoing releases and where they are and at which stage they are and we can also see the ESD RTM media.  
Like like the realistic at the that we were looking at and we can see that it got published to internal audience and signed off on and then it got published to retake. So movies 6 publishing is integrated through all our systems it's it's visible in track it and it's integrated with the AWE but not fully automated yet. And on the one pub side it's fully automated meaning that.  
The creation of the updates and the processing of those requests and everything happens automatically. We don't need to do anything.  
Umm I can dive more into those operations, but let me know like which which parts would be more interesting for you.  
Uh, I see the comments about the access issues. Uh, I I can look into that after the meeting and and we can work to give you access to.  
To those portals.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 21:21  
Do you know I had a question.

 **Dina Helal** 21:23  
Yes.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 21:24  
Actually, so these publishing.  
Things these two have the automatically approval capabilities, right? So what does the third one?  
The golden yes.

 **Dina Helal** 21:36  
This one.  
OK, so publishing to test audience. So publishing for internal validation inside it has automatic approval because the update will not be available for the validation teams to pull it from the dogfood. More server without it being approved. So that's why it has auto approve embedded inside it on our side. But publishing to retail doesn't have auto approve embedded in it. So publish to retail only.  
Makes the updates submitted to move, but they're not available for anyone to download until approval is kicked off in a separate operation.  
So this is by design that that publishing to test audience has auto approve but publishing to retail doesn't.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 22:27  
And these approvals are given by the Adams.

 **Dina Helal** 22:31  
Yes. Yes, exactly, yes. So they are the ones that follow up with the communication between US and the validation teams and like because there are different kinds of validation that happens. So, so they have an overview of what is the validation status so far of this.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 22:32  
OK.

 **Dina Helal** 22:51  
Of the skellies, so they give the final approval and then.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 22:58  
And what does the this dog food service? I'm sorry, I.

 **Dina Helal** 23:03  
That's OK. Go ahead.  
Sorry you didn't get the question.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 23:09  
Don't tell that you're talking about what is that?

 **Dina Helal** 23:13  
So this is.  
The internal move server.  
That's it. Like so.  
It's like pop sweets. So when we publish to pop suites, we are publishing to this internal testing environment for validation. So dogfood moves. Server is also the the internal move server where we publish for validation. So no, no one from outside will be able to pull any updates from it. This is our Moore testing environment basically and it has a different endpoint than the live 1.  
So in when we publish to test audience, we are publishing to this internal.  
And point, let me see if we have it mentioned here. So you see this is the publishing endpoint and it has you see dog food in it and this is a different endpoint than when we publish to agree tails.  
So this one doesn't have more in it, so it's a different server. Basically one is used internally and one is the the one that enterprise customers get to pull the updates from.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 24:38  
Thank you.

 **Dina Helal** 24:39  
Sure.  
So the way, sorry, go ahead. Go ahead.

 **Bhaskar Verma** 24:47  
Do you know what are the?  
Yeah, I was saying that. What are the contents of ESD bundle? Are there other files or or things also that are added in it?

 **Dina Helal** 25:00  
Yes, definitely. So let me show you.  
Umm, so we have something called publishing XML and this is the the this is what we publish to the dogfood move or the most server. And what this publishing XML has is.  
All the targeting information about this update and the files included in it.  
Let me.  
Find like one of the ESD.  
Updates I think I had one recently.  
No, not here. Let me let me find you one and and we can take a look at the eczema.  
So we store all our.  
Data in Cosmos DB and in a storage accounts and so the pub XML in the end are uploaded to to a storage account and then pulled from there and published to the to the move server.  
So among the data that this update has is.  
Is the the.  
The yes, the location.  
So that the path to the ESD file and the Windows Update box. So this is the content that gets published in the end, in addition to to all the metadata associated with it about who can install it, and this is currently an on Prem file location but this will change while it gets uploaded to the cloud as well and we're working to to use the the cloud.  
You are I instead of the on Prem location.  
Let me see if we can access this now.  
And Creek and.

 **Bhaskar Verma** 27:52  
Currently it is we use the file share path where we keep the ESD.

 **Dina Helal** 27:56  
Yes, yes, currently currently the the Media creation publishes that to to to two locations the on Prem and the and the and the uploads it to Azure as well. But we are getting it only from the on Prem location.  
For what?  
So this was published a while ago, so it it may have moved to A to another location or not available on this check anymore, which is what it looks like. But if we get a more recent one.  
We can.  
Find it.  
So let's see.  
And and the way the communication happens between.  
Umm, AWE and one pub is through a service bus. So.  
AWE calls.  
A service that puts her message in the service bus and then we have a logic apps that are listening for.  
And for those messages and and then they consume those messages and those messages would have.  
Umm.  
I request type, so publish or approve and and they would have the.  
2nd and they would have the information on this.  
Egg quests.  
But I can I can share with you later the location of an existing.

 **Bhaskar Verma** 30:27  
So the cloud share path location is the cosmos DB1.

 **Dina Helal** 30:29  
Actually.  
No, that's something else. So let me show you here what our build metadata looks like.  
So for each file we have this path and we have this.  
Umm this Uri, this is where it gets uploaded and this is a copy of of that other file. Those are exactly the same.  
We are currently using the on Prem location but we are planning to move to to this.  
Umm, what was blocking us from this is that pubsuites don't support publishing with the clouds Uri yet. They only support publishing with the UNC file path.  
But this is hopefully changing soon on their side and on our side as well.  
Maybe I don't have access to view those anyway.  
So yeah. So for example on our side, when we get the published request, we have a service that using those.  
This build ID.  
Connects with an external service and gets all of this.  
Build information which has, like all the ESD's for all the languages and all the different architectures and additions and then creates all the different updates and then each update puts the path to the corresponding yes the file.

 **Kush Mishra** 33:08  
And I've given you file location. Can you try that?

 **Dina Helal** 33:09  
Yeah, I saw it. OK, let's try that.  
So we didn't get the full pass.  
Uh. Maybe because I'm not.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 33:45  
I can access this.

 **Dina Helal** 33:45  
OK, so, so OK, OK. I'm not sure why I can, maybe because I'm not in the office, I'm not.  
The connected. So anyway, thanks Kush for sharing that. So yeah, this is this is where we get the files from currently.  
And then.  
We our services builds this XML file and what this XML file has is.  
Umm something called?  
A at an update and the child update. So the one on top is the child and the second one is like the pageant and it has like the update title, the time to go live.  
And it has something called prerequisites.  
And those prerequisites are those guids and each GUID.  
Represents something so one of them represents like the architecture go with the other one. They represents the edition. So each guide of those has.  
Logic embedded into it.  
Uh, so if we search for this one, for example on Pubsuite Explorer, so all of those exist on on Pubsuite Explorer.  
We should see. So. So this one is hardware floor enforcement detectoid. So this is specific to to Windows 11.  
And it has additional logic that specifies.  
And the the minimum build version that the machines have to be on in order to to install Windows 11 for example. We don't add this detectoid or this GUID in the prerequisites of Windows 10 updates because it's different like for Windows 11, it's not applicable to all the machines, so that's why we have this like additional logic.  
And if we find some another one.  
Might be architecture or product specific, so this is for example the architecture detectoid for each architecture we have a GUID that represents.  
What does it mean for the machine to be on this architecture?  
And so on. So we have. So what we do on our side is we have something called admin data we have like admin data is like rules that are embedded in our system. And whenever we get the request to publish a certain product, we go and read those rules and create bundles based on those rules. So one of the rules that would be in our database for example for ESD's is that.  
X86 is supported for Windows 10 products and not supported for Windows 11, so this is how we know not to create any X86 updates.  
It it has like which editions are supported and and like which languages and and so on.  
So all of this is already embedded in our system, and once we get Aga quest to A to publish something.  
We read this data which we call admin data and and then we.  
So like we so this is one of our logic apps and what this logic app is doing in the end is it's.  
Reading the admin data, validating it, getting the the files, the ESD is the builds and creating the bundles and then.  
It publishes those so it loops on the 114 bundles and submits them to to dogfood more or to live more, depending on where we are publishing this.  
And you see here it's checking if if we have anything to auto approve. In this case it will not auto approve because this one was publishing to retail. So we will not have the the O2 approve option for it.  
If, if we were publishing to test audience then this logic app would have auto approved those for example.  
And in this logic app we are calling other logic apps so in in this loop where we are looping on the updates we are calling another logic app which submits to to the appropriate endpoint and.  
And this this is how the flow is going on on on the one pub side, I didn't dive to too deep into it because there are a lot more details. So depending on what you want, we can go deeper or not.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 39:00  
Yeah, I have one question. So is it the same example that you used to while submitting it to pubsuite while calling through client agent?

 **Dina Helal** 39:10  
Yes, it is the same pub XML, yes.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 39:13  
I know.

 **Dina Helal** 39:14  
Umm.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 39:15  
That incident question.  
Do we have a pretty mad logic given by Pubsuite team for generating 6 Mr?  
How is different?

 **Dina Helal** 39:25  
Sorry, do we?  
OK.  
OK.  
So we have a service that.  
Knows how to construct this XML with the guides like elements and the guide data that pubsuite would accept.  
And for yes, this we we submit to to more directly. So we submit to dogfood more or to live more directly. But we do submit to we have an ESD pop suite that we can submit to for for internal validation in our dev and PPE Invironment. So it is the same pub XML format, it is the same structure and.  
It can be submitted to pop suite for validation and in the end in in production we submitted to so it it is the same like the the same pub XML. We know what it looks like because we we have a service that is create that that creates this axiom L in the right format and we have a service in shared services and.  
And actually show you this.  
So we're remember we we probably mentioned that in in an in a previous session that we have a shared services cluster and a main cluster. The main cluster has like the movie 6 service fabric application and the ESD service fabric application. And then we have the shared services cluster which can be which is used by different themes and different pipelines. And one of the services in this shared services plus deck is more publishing service.  
And.  
And the small publishing service internally uses the.  
The client agent DLL to submit A2 pop suite or to.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 41:50  
Thank you.

 **Dina Helal** 41:51  
Sure.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 41:56  
So they'll be they actually.  
Does that ingest chapter ESD the whole the huge size file plus?  
Do that or it is just that we give that article the.  
Other movies.

 **Dina Helal** 42:13  
Sorry, I didn't catch that. Sorry. Can you say again?

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 42:17  
The question is later when we say we published to move or dogfood move, or like giving the event C back.  
That's the moon a bit. I can't take the pattern, consume the files directly or or do we write that logic right in terms of public?

 **Dina Helal** 42:35  
Uh, so on on the move service side, they they consume it, we published with the UNC file path on our side and at some point they they consume it and they.  
Uh, I think I remember when I when I was downloading some of those files from the move server directly. I think this changes to to something else but on our side we we publish with the UNC file path.  
So if you go back to AWE, you see we have one play here and what this?  
What this play means? It means that this is a specific payload, so this is the the payload that is coming from the CPT that we published. So let's say we found a bug or an issue while validating this. So what happens is we can go on AWE, start a new play, it will have a higher version version 1.1. So this will be a new payload that we are publishing coming from a new build.  
And what would happen automatically on on our side if we get such request is that we will see if this is.  
A new build that is different from the previous one that we published. Then we have to expire the previous bundles. Basically take them down, they they won't exist anymore and then create new ones. So that's why we have a step here for example called get bundles to expire. So what happens here is that the checks is is the new player, is the new churn, has a different build than the one we published before. If it's different then we have to.  
Kill the previous updates, expire them and then create new ones and and publish them. If it's the same build. But let's say we are changing the time to go live, so something happened and we are pushing the TJ out or something. Then it's the same files, the same build, the same everything but ttgl is different then we will not expire. We will use the same bundles but on those bundles on those updates the time to go live will be updated.  
So this is part of the logic on the one pub site to determine if we are getting a republish request. What kind of request is is this? Is it updating metadata or is it a completely different build so we need to to change the updates?  
Umm.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 45:46  
So what happens when we create a new play? Does it?  
Create a new release ticket.

 **Dina Helal** 45:54  
No, it's within the same release ticket, so it will be like the same release ticket ID. It will just have an additional section here called version 1.1 for example, and it's a new churn like also in trackit, when we expand like that, we can see how many versions were there.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 46:06  
Please.  
Ohh.  
Thanks.

 **Dina Helal** 46:19  
Of this built, so yeah, it's a new belt and on on the one pub side it will appear as a new published request. So I can have this publish test and then have another publish test which is like overriding the previous one and then published to retail.  
So we can have as many publish requests as we want.  
Until we have the the final one.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 46:46  
And in which case would the bill be different for a new play?

 **Dina Helal** 46:52  
And if we find a a bug during validation then.  
And then a new media will be generated. So if if we find an issue and and and we have to create like the media team has to generate a new build, then they have to to share the new build with us. So this will be different and and then this will be a new play and a new payload that that we are publishing.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 47:26  
If there are new, no new check-ins after the last play, then there is no new build.

 **Dina Helal** 47:31  
Yes, exactly, yes.

 **Bhaskar Verma** 47:37  
So, Dina, can we say that a, a, a new place equal to a new churn?

 **Dina Helal** 47:42  
Yes, yes.  
And yeah, as Helena mentioned, she will cover AWE next week and Kush showed also an example of what it looks like to have different churns on on AWE.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 48:03  
So.  
The final term is when the sign up is done by the respective testing team model.  
Who? Who makes that call that this is the final check the daughter or?

 **Dina Helal** 48:21  
So yes, this come from the the big list. So what happens is.  
It it we're not supposed to to have a lot of churns. So what happens is this is the the the 9B built and.  
And supposedly it it should be like it should be the right. But sometimes we have to republish if we have to change something on the targeting logic on our side. So what the validation team does they like they they check that everything is fine with with the build itself and with the targeting info that we have. If we have to have a new build then we'll do that. But this is not very common.  
And you.  
There isn't like a standard way of communication. It could be a on the CPT itself. So for example I I generated the bundles for test audience and and I I put the IDs here and then I generated the bundles for for retail and I put the ID here and I share the CPT with the validation team and and they sign off whether it's on e-mail that that the Rams are on or on the teams chat it's.  
There isn't like 1 standard way to do it.  
But the arms and the PM's are always on on top of it to to know that that everything is fine and that everyone signed off on it to proceed with the final approval that they kick off from the packet.  
So any other questions?  
I will share with you the slide deck. It had some more details about how our system works internally.  
Ohh, reach out to me if you have any questions or we can cover it in in another session if you're interested.  
But basically we're we're showing here how how we what happens once we get the call from AWE till we publish the to the more server.  
And and how we are using the logic apps and.  
And all of that.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 51:19  
Here.

 **Dina Helal** 51:32  
Right then, I guess we can stop at this point and please reach out if if you have any follow up questions and and we can do more sessions if if you want to dive deeper into this.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 51:45  
OK.

 **Jelena Maksimenko (CPL Solutions)** 51:52  
Thank you, Dina.

 **Kush Mishra** 51:52  
Thanks Anna.

 **Dina Helal** 51:53  
Thank you.

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 51:54  
Can you give me can you?

 **Bhaskar Verma** 51:55  
Thanks, Dina.

 **AB# Mishra** 51:56  
Thank you.

 **Dina Helal** 51:57  
Thanks. Thanks, bye.

 **AB# Mishra** left the meeting

 **Lacastan Moodley** left the meeting

 **Halyna Hladkivska** left the meeting

 **Pinar Sen** left the meeting

 **Jelena Maksimenko (CPL Solutions)** left the meeting

 **Bhaskar Verma** left the meeting

 **Kush Mishra** left the meeting

 **Mustafa Mizrak** left the meeting

 **Conf Room HYD -CAMPUS 2/26 (13) Boreal Priority** 52:09  
It doesn't turn off.  
And Andrew?

 **Dina Helal** left the meeting

 **Kruti Kallola Mohanta** stopped transcription