**Part 9 (AWE)**

August 31, 2023, 3:36PM

23m 3s

 **Mohit Gupta** 0:05  
Yeah.  
No, it's thanks for asking.

 **Tavishi Gupta** 0:10  
OK, so these are the API or like the contract between a WE&S, awe sends the intent which is which we call the AW media creation request on a service bus on AW service bus.  
So and then when once SMF receives it, they send an acknowledge message to AWE.  
And then so this is the acknowledge acknowledge event from SMF, then they do the media creation and then they send the AWE intent completion notification which is basically umm then the which is basically what we call the sign off and on the service bus we read we did you it, we read it, we start the play this is the publishing goal I process that either we start and then we send then we send and then we'll see you on the next slide but then we'll see how we communicate with one pub but on the same service bus one thing is important to note is that we we actually like after like the release is gone live we actually send a go live notification on the same service bus to SMF and prod to umm that's space as well.  
So they use this live notification to to basically mark the media life on their end.  
Tomorrow.  
OK, this is the AWE one pub contract in which when the so this this this starts when AWE starts to play.  
So we do ingest, publish, approve, go live.  
All of these are all of these are API cards that are gone through that that goes through.  
This this ipublish interface is coming through one pub next Gen and media team as well, like Media Publishing as well.  
And then this basically, but because it's the same, we use a type.  
So in here this the channel is media so so this ipublish like it reads it, it sends to the right service bus and then ridicule it.  
Ohh, on one pub services.  
Umm, we start a logic apps and the recent after the logic app like after each step is successful or fair we send a notification for the success and failure.  
AWE reads this and marks its activity either in success stage or in field stage.  
And then the so if if, if, when we read it we complete the activity, if we if we just fail if the status is failure we fail the activity.  
This is a little.  
It's a little bit of uh this, there's a lot of competence here.  
So Mohit, you have your hand up I see.

 **Mohit Gupta** 3:37  
Yeah.  
On the previous slide, I have one question.

 **Tavishi Gupta** 3:41  
Yeah.

 **Mohit Gupta** 3:42  
So in the very end you said the release info is dequeued by one pub services, which triggered the logic apps.

 **Tavishi Gupta** 3:53  
That's right.

 **Mohit Gupta** 3:53  
Is that?  
So one pub services trigger the logic app and then logic app again triggers the.  
What we call one pub services.

 **Tavishi Gupta** 4:08  
No.  
So, OK, what do you mean one per services here?  
Because this is actually media publishing.

 **Mohit Gupta** 4:17  
Yeah.  
Media publishing.  
Yes, the code base in the one pub cloud repo.

 **Tavishi Gupta** 4:23  
Right.  
So Media Publishing is triggered here.  
So India's publish approve go live.  
These are all logic apps in that in our in our umm, in our service in our code right which triggers actual ingestion, which is basically the release documents are actually created there publishes what it actually goes to whatever endpoint it needs to go for containers.  
It's the MCR for VHD.  
It's the marketplace.  
Then approval is like when the items are proof we copy or we flip the flag right.  
Sometimes we copy it to a public location and go live is we actually is in containers like.

 **Mohit Gupta** 5:04  
Maybe you too much.  
He.

 **Tavishi Gupta** 5:11  
I think Jacob, you can grab me if I'm wrong.  
Containers.  
We actually flip the flag and that's how we we just say go like, yes, gone like in VHD, we what do we do in VHD, we actually we also mark them as life, Yep.

 **Mohit Gupta** 5:27  
Calling the API call one call us APPF.  
I'm not wrong.

 **Tavishi Gupta** 5:33  
In marketplace, I think OK, yeah.

 **Mohit Gupta** 5:35  
In marketplace, yeah.

 **Tavishi Gupta** 5:39  
And then and then, yeah.  
And then on the same service but and then this notification through this logic app through this logic app like has either something happened right exceptions happened or incident happened, we will send us a failure notification on one pub.  
Also, you'll need to believe this is through a service bus.  
Oh sorry, this is this is through a service bus and then through that I publish EPIN.  
Having two Q will be basically say that the IT has completed or failed.  
We are all we are listening to these notifications on AWE.  
I'm not sure if I answered your question more than.

 **Mohit Gupta** 6:27  
Yeah.  
Yep, let's, let's go ahead.

 **Tavishi Gupta** 6:30  
OK.  
Yeah, this is a lot of competence here.  
I'll try to explain.  
So we start with this every, umm, 35 days in advance we create realistic again.  
This is an automated process.  
We create release tickets which basically is a VSO.  
Umm we use release release manager is basically the service on AWE that actually that that is responsible for creating release release ticket after this release is release ticket is created.  
Look, I think around past Tuesday we will create 10, we release ticket for example, after that on those release tickets, we will start reading, we weekly completes which is basically the packages for that particular app or for that ohm pillow for is or like sorry the payload has create being created for publishing or that update type.  
So we listen and all these are shown on tracking.  
So we we listen to the BVT complete and so and then this is how non median AWE uh media are hooked up basically.  
So we just release release manager listens to the beauty.  
Complete it kicks off a play.  
This is the non media play this step to non media and the service which is responsible for that is really the automation.  
Then it kicks off the activity of the play.  
Play kicks off activities that is responsibility of the activity controller and then this basically goes to and then we kick off activities.  
It's basically ingest, publish, approve, go live on the non media side.  
So this is non media right now and they will see how the same thing be what happens is like one this non media play kicks off and then this media kicks off which is the BVD trigger here.  
So the media orchestrator is the main me and service that that basically takes care of a media creation and play kick off.  
So we read so on media appreciator we read the media config first, which basically tells us whether an intent is applicable or not.  
If it if we need that the intent is applicable, we start creating intent for it which means from from one pub we get the get the media list and from a play runtime we need the payload.  
We use this two information.  
These two pieces of information we save we we save it in our records, in the Media orchestrator runtime and then we send resend.  
Umm we send the intent 2 etc.  
The media.  
When, when?  
When in the intent, when the media is created, we get the sign off here media officials orchestrator listens to it.  
We send then we start a play.  
Do and.  
Then the play started using release automation.  
The release automation kicks off activity.  
Umm.  
And then publishing starts for ingest, publish, approve cooler.  
We send it to one pub.  
One pub does the.  
Ohh does the actual I ingestion publishing on 2 pages endpoints approval and go live and this is a.  
These these this is kind of like a look on what are all of that services involved and all of this information is.  
So all of this information is involved is showed to track it for for I'm with you or I am tracking and auditing any questions here.

 **Aravind Siddoju** 10:58  
No tavishi one question.  
So this picture is very clear I have.  
I'm little curious, curious from some time now on so.  
Basically, when we process the request on the onepub side, our close to a a notification or put a message in AWE service first as a callback informing AWE that activity are the media operation is completed.  
So how how would AWE consume that and proceed to the next activity for the next stage?

 **Tavishi Gupta** 11:32  
So the these are callbacks, right?  
So it's waiting on it.

 **Aravind Siddoju** 11:38  
So all activities are basically that adapters which waits on the callbacks.

 **Tavishi Gupta** 11:43  
Correct.  
They hooked up through the publishing adapter.

 **Aravind Siddoju** 11:46  
And.

 **Tavishi Gupta** 11:48  
That's right.

 **Aravind Siddoju** 11:48  
Umm, like and once they receive the call back which the waits on.  
So do do we have any specified logic in AWE to invoke the next activity or is it a sequential execution of the activities that we mentioned in the play information the play JSON?

 **Leopoldo Estrada Vargas** joined the meeting

 **Tavishi Gupta** 12:09  
It is the sequential like so there is workflows right which does.  
It's basically a flow chart.  
This can happen at like some activities happen in parallels and had activities happen sequentially.

 **Aravind Siddoju** 12:18  
Umm.  
So it's kind of a dependency graph kind of thing.

 **Tavishi Gupta** 12:24  
So.  
Correct.  
Exactly, exactly.  
That's what it is.

 **Eduardo Toraya** joined the meeting

 **Tavishi Gupta** 12:29  
And then and then that logic basically tells us like that dependency logic that tells us whether we need to start the activity sequentially after finishing the previous activity.  
Or we can say we can start activity A&B together.  
In parallel as well.

 **Aravind Siddoju** 12:47  
OK.

 **Tavishi Gupta** 12:48  
But then the next sequential activity you would not start until leads to activity like completed and we get a successful it.  
If you get a feel we are not going to start another actor.

 **Aravind Siddoju** 12:59  
OK.  
And we maintain the dependency information again in the play, Jason, yes.

 **Tavishi Gupta** 13:04  
No, that is.  
Umm that is maintained and workflows.

 **Aravind Siddoju** 13:11  
OK.  
OK.

 **Tavishi Gupta** 13:16  
But that is like in.  
If you look at the, if you look at the.  
It worked closely.  
You will see workflows and activities and so it just it is actually maintained in the workflows project.

 **Aravind Siddoju** 13:23  
Umm.  
OK.

 **Tavishi Gupta** 13:35  
Any other questions here?

 **Aravind Siddoju** 13:35  
So, yeah, I'll, I'll take it.  
Thank you.

 **Tavishi Gupta** 13:40  
Good.  
So we just like going back to what is an intent, what does it basically do the intent it triggers media creation.  
It includes WSD non artifact no or WSD artifacts that are known to AWE.  
This is important because there's certain things that are not like we we we don't we we like that information doesn't go through W.  
So actually, media team add that information would gets it from somewhere else today.  
Umm.  
But is it example of this?  
It it used to be .net actually.  
One example was .net, but now AWE like .net AWE knows about.net.  
So we have .net releases, sonow.net information.  
We have enabled.net through AWE, but there's certain customizations with that.  
AWE doesn't know about, so they add those customizations to.  
To the media, do the payload to create that different type of media.  
So this is an example of what?  
Uh, uh.  
What we we like when, when do we trigger?  
What?  
What do we do actually?  
So then I said BVT, we also do cat completed.  
We also send intents on CAT completed.  
Why do we send two and then?  
This is the reason sometimes they SMF is like hey, why did you send me duplicate even though they have duplicate uh logic detection which means like if there are multiple in 10 sent to them they will only pick up the the first one they are not going to start triggering media on.  
On a duplicated payload, but we do this because sometimes cat can get cancer.  
That means most 90% of the time it's not going to get cancer.  
But sometimes, like the package can get cancelled.  
That means that we don't want to send that information anymore to umm.  
I do SMF because that payload doesn't exist anymore, so we always like send a CAC.  
Uh, we also like.  
Ohh trigger on CAT completed and on this is this is an anomaly to the previous.  
Ohh sorry previous previous two which is on Patch Tuesday.  
We actually wait for the office team to send us the office payload which is which, which is what is used for office and CPC media, and then that's what we office and CPC media always goes on.  
Intern intern always goes on patch Tuesday.  
This is basically what an intent looks like.  
It is.  
It is basic, it is.  
It is a composition of media list which is this guy and the payload and this case. So LLC.  
SSU setup safe OS EKB whenever it is applicable office or or office for only office and CPT for Sun Valley products we send Uno as well for hot, hot patch product we send for no hot batch hotpatch LCU SFSU safe OS DU and setup DU.  
That is.  
Basically, the end of the AWE side and let me know if you are then questions on this like could the I can see your hand up.

 **Kruti Kallola Mohanta (WSD)** 17:27  
Yes, my question is around what are the frequently tools used in AWE area.  
So at least one I think is required is like let us say we are trying to do local testing or in PPE testing then for me pub we have a token generator tool where we just click a button and we get a token and we're able to test the API.  
Similarly for AWE testing.  
Is there any token generator tools or how to create tokens?

 **Tavishi Gupta** 17:54  
I don't use it token generator, but I I I think maybe there used to be a script long time back which I which I for me to broken.  
I don't.  
I can't good to work it, so I use postman actually and I use you the way I do it at the longer way I actually look at the.  
Because it's the release I release, right?  
So I really use the release automation portal like on.  
If you look at the production or sorry not production, uh locally and P, uh, keyword, it has all the secrets and everything saved.  
So I use the port post Mac.  
I use postman to actually create a A, create a, create a authenticator token.  
Then I plug it into my API, then make it that way.  
It's not.  
I mean, I think Paulo has.  
Made it easier on the one pub side by creating that token generator.  
There used to be a script and I think maybe people use it still.  
Umm.  
But I think for me, for me it never works.  
For me it's broken.  
So I used this longer way to do API calls.

 **Kruti Kallola Mohanta (WSD)** 19:14  
OK.  
So yeah, like we have the from the keyboard to take the secrets and use those instead in postman or invoking calls something.

 **Tavishi Gupta** 19:25  
Right, that's what I that's how I do.

 **Kruti Kallola Mohanta (WSD)** 19:26  
So any of that either means frequent tools that you use in AWE that we need to be aware of, like any frequently used tools.

 **Tavishi Gupta** 19:32  
I think are you think the I think I'm not sure if there's nobody I probably know about it.  
The service bus, I think there are two ways, yes.

 **Kruti Kallola Mohanta (WSD)** 19:40  
Because I just was explaining.

 **Tavishi Gupta** 19:42  
So a service bus, either you can I I have the application on my end, but you can go to the portal Azure portal and see the service bus.  
You can you can so sometimes like.  
Ohm sometimes like either either like either we have in testing mode or something like really bad happened like I don't know last what was the thing last time there was something.  
Something happened that I have to send manual intents because there was something had broken.  
So we send we we we we send service bus.  
Intense.  
Uh, through.  
I'm I mean we send interns to those service bus.  
So either use the Azure portal to do it and I use the I.  
I personally use an application.

 **Kruti Kallola Mohanta (WSD)** 20:37  
And yes, there's.

 **Tavishi Gupta** 20:39  
Those are the two important things.

 **Kruti Kallola Mohanta (WSD)** 20:41  
OK.

 **Tavishi Gupta** 20:42  
And then you guys, you, you and then there's obviously there's all the AWE portal we use all all the time to basically check the the where are we in the umm if to basically check the things or uh the activity state or the media said.

 **Kruti Kallola Mohanta (WSD)** 20:51  
And.  
But it looks like a AWE airport and P or phone blocker and service was explorer.  
They are the highly frequently used to explore testing and all packages in there.

 **Tavishi Gupta** 21:19  
Right.  
And then when once if I'm picked us so like I said in the presentation that two important tables, right, there's a media orchestrator conflict table and media orchestrator runtime.  
So I will because every orchestrator runtime always saves a record of all the intents.  
So if tomorrow, like it has happened in the past like I SMF was like, hey, we have not received this.  
So you can just look at our tables and you say hey, it shows that it was sent.  
So then then they go and check their side.

 **Kruti Kallola Mohanta (WSD)** 22:02  
And uh, PME.  
Like in AWE.  
What is that PM like?  
Everything is married to PME or like few things are still in MSFT.

 **Tavishi Gupta** 22:12  
You PME P PB production are all in PPE.  
PME locally you have like you can do anything.

 **Kruti Kallola Mohanta (WSD)** 22:21  
And.  
There's some from my side.  
Thanks. Noisy.

 **Tavishi Gupta** 22:33  
Including.

 **Mohit Gupta** 22:46  
I think you're on time for the two PR walkthroughs.  
Who wants to go first?  
Or Arvind or Bhaskar.  
And should we stop recording?

 **Aravind Siddoju** 22:57  
Now we can stop recording.

 **Bhaskar Verma** 22:58  
Yeah, I can.  
Yeah, I can go first.  
I'll stop the recording.

 **Bhaskar Verma** stopped transcription