

8

00:00:23.880 --> 00:00:34.379

Brian Ruf (EZD): Well welcome everybody uh I'm Brian rough. You just heard Chris Robey. Speak up a moment ago, Chris. You and just give yourself a quick introduction.

9

00:00:34.800 --> 00:00:50.079

Chris Roblee: Ah, yeah, i'm. I'm work. I ah consultant here at easy dynamics. I work with Brian's team. Um. Assisting with various ah product development efforts. Ah! Including the Oscar Registry, that we'll be working on here.

10

00:00:51.570 --> 00:01:08.490

Brian Ruf (EZD): Thank you, Chris. Yes, and I i'm Brian rough. At one time I was part of the this Oscar team as a contractor, was involved in creating the oscill standard. Now i'm at easy dynamics where i'm building tools, um leading efforts to build tools

11

00:01:08.500 --> 00:01:14.899

Brian Ruf (EZD): that that will hopefully further the cyber security industry with with automation

12

00:01:14.910 --> 00:01:28.649

Brian Ruf (EZD): um and these capabilities that we'll be talking about Here were briefed in January as part of a mid-atlantic oscill community meetup that we facilitate

13

00:01:29.150 --> 00:01:46.339

Brian Ruf (EZD): the ah that meet up meets every um three to five months in the Dc. Metro area. So you'll see some branding related to that on these slides today and the same email address We're going to give you for information about the slides is also one hundred and fifty,

14

00:01:46.350 --> 00:01:50.299

Brian Ruf (EZD): where you can find out more about the meet up if you're interested in participating.

15

00:01:51.260 --> 00:01:57.849

Brian Ruf (EZD): That is a open to anybody who's developing tools for oscill trying to implement oscill in any way.

16

00:02:00.050 --> 00:02:02.070

Brian Ruf (EZD): Let's go ahead and jump right in.

17

00:02:02.080 --> 00:02:27.770

Brian Ruf (EZD): We have a few things we want to tell you about today. These are all um features or capabilities that we're we've been working on. We uh with the idea of making them available to the oscill community. Um, and in hopes of ultimately having the oscill community collaborate with us on these items, and that's part of why we are, uh presenting them under the auspices of the meet up group.

18

00:02:27.780 --> 00:02:37.619

Brian Ruf (EZD): Um rather than as ah representatives of these dynamics, and where we view ourselves as more the facilitators of these community capabilities right now.

19

00:02:37.980 --> 00:02:46.020

Brian Ruf (EZD): So we'll tell you a little bit about Oscar Io, which is intended to be a hub for oscal information.

20

00:02:46.030 --> 00:02:57.849

Brian Ruf (EZD): We're going to introduce a an Osc out Content registry which will be open to the public Ah, along with that a no-scale content viewer those two things go hand in hand. We'll have a little demo of those two.

21

00:02:57.860 --> 00:03:10.389

Brian Ruf (EZD): We'll talk a little bit about a rest, open api specification that we think the community surely needs. There's been a few Api um rest Api discussions

22

00:03:10.400 --> 00:03:23.670

Brian Ruf (EZD): around over the last few years, so this is our contribution. Um! And then we'll, we'll get into just some ah additional information and um resources that are available to to you as community members.

23

00:03:25.760 --> 00:03:40.790

Brian Ruf (EZD): Um, please be aware, Chris is Michaela are both monitoring the chat. So while we're going along, if you have questions, if you could drop them into the chat, we will have an opportunity at the end, as Mikayla said, for open questions.

24

00:03:40.800 --> 00:03:53.439

Brian Ruf (EZD): Um, So you know as you go, please let me keep my flow, or as I go, let me keep my flow drop questions in chat, or be prepared to ask them verbally later. In the presentation

25

00:03:54.420 --> 00:04:07.449

Brian Ruf (EZD): Oscar Io is intended to be a portal kind of a one-stop shop for people trying to get started in the Oscow community or trying to find more resources. We have deployed the site.

26

00:04:07.480 --> 00:04:13.579

Brian Ruf (EZD): It's out there. It still has a lot of growing to do. The even the

27

00:04:13.640 --> 00:04:31.189

Brian Ruf (EZD): um github repo for managing the site is a public repo so that others can contribute Um, basically it's a place where we can get events listed. We can start to list any known tools and

28

00:04:31.200 --> 00:04:43.949

Brian Ruf (EZD): communication channels. Some of these are implemented today. Some of them are are planned as you're seeing here. And this is also where you can, where you will eventually link out to the Content Registry and the viewer.

29

00:04:44.460 --> 00:04:50.939

Brian Ruf (EZD): I'm gonna just give a quick switch over here and make sure I get the right tab

30

00:04:53.790 --> 00:04:55.890

Brian Ruf (EZD): stealing Thunder! This

31

00:04:57.080 --> 00:05:01.859

Brian Ruf (EZD): This is Oscar Io. Actually, this is what the landing page looks like

32

00:05:06.510 --> 00:05:18.639

Brian Ruf (EZD): it's again. It's a start. It's out here. We do have so far a an events page and a tools page,

33

00:05:19.760 --> 00:05:36.460

Brian Ruf (EZD): and we're looking for emit right now that this the I'm. Sorry stammering right now. The idea is that you can send requests over email to have things listed on either of these pages, but eventually there'll be a self service portal.

34

00:05:40.800 --> 00:05:43.320

Brian Ruf (EZD): Get back with the briefing here.

35

00:05:45.500 --> 00:05:47.420

Brian Ruf (EZD): Why, it went out of

36

00:05:47.820 --> 00:05:48.920

cash,

37

00:05:53.860 --> 00:05:58.689

Brian Ruf (EZD): as I mentioned, we're um. You know we have a tools list

38

00:05:58.980 --> 00:06:09.579

Brian Ruf (EZD): today. Communication channels coming soon. Some of the things you're going to see on the next few slides are proposed criteria for inclusion

39

00:06:09.590 --> 00:06:21.749

Brian Ruf (EZD): mit ctl and um, and some of the proposed details we believe, should be listed there. These are things that we're looking for. Community input on. We want the community to shape this to be something that makes sense for everybody one hundred and fifty.

40

00:06:23.730 --> 00:06:28.550

Brian Ruf (EZD): So we have a drafted criteria. That

41

00:06:28.640 --> 00:06:57.119

Brian Ruf (EZD): that's part of why we're keeping the process manual today as we get clarification on the criteria. Our intention is to create that self-service portal. But for today this Moscow. I O sorry, Oscar. At Pascal, Io email Address: Um, Hopefully, that's easy to remember. You'll see it several times in this presentation. Um, really everything you see in this presentation. That's the email address to use, to get more information or to get involved.

42

00:07:00.870 --> 00:07:06.510

Brian Ruf (EZD): So for tools where the criteria we're considering

43

00:07:06.520 --> 00:07:25.690

Brian Ruf (EZD): is that to be listed as a tool in the list that you have to be able to support any version of Moscow. Starting from one dot o or later. Um, you don't have to support the latest version. We recognize that there will be drift. Um, but you have to support a valid version of Moscow.

44

00:07:25.700 --> 00:07:44.650

Brian Ruf (EZD): Um in some way whether and tools could be a library, it could be open source. Um! It could be a a um, a for profit closed source. Um license tool, where there's no limit on the type of tool, only that it supports Moscow.

45

00:07:45.580 --> 00:08:02.090

Brian Ruf (EZD): Some of the details, we believe, should be listed. Ah, the name a description What's on scale versions you support, and you know some of the details about how how you would be contacted as the tool owner. What type of license you're offering one hundred and

46

00:08:02.100 --> 00:08:07.829

Brian Ruf (EZD): we This is an area where we want your feedback. Is this the right criteria?

47

00:08:08.260 --> 00:08:19.209

Brian Ruf (EZD): Is this a a tool like. Is this the right detail for tool entries? So anything different? Anybody thinks they we should see there.

48

00:08:20.210 --> 00:08:30.280

Brian Ruf (EZD): And again. Please drop information into the chat, said email to Oscar at Oscar. If you have input on the tools,

49

00:08:34.080 --> 00:08:53.429

Brian Ruf (EZD): it's very similar for communications channels. So the idea here is that we would list things like the the nest um mailing list and the getter community for Oscar the linkedin uh ascal community, the even the mid Atlantic Oscal, meet up.

50

00:08:53.440 --> 00:09:08.729

Brian Ruf (EZD): Ah, all of those things would be listed here. If there's any other communication channels that anybody is hosting for any reason related to Oscar, we want to be able to list it at Oscar that I owe so people can find it.

51

00:09:08.790 --> 00:09:15.189

Brian Ruf (EZD): Uh again. We um. The only thing we really think is important is that the

52

00:09:15.220 --> 00:09:32.330

Brian Ruf (EZD): Communications Channel must be related to Oscill in some way whether that's development, implementation, adoption doesn't matter how it's A. Related to that. But that's the only requirement. Now we asked this in January, the meet up events about whether this should be

53

00:09:32.420 --> 00:09:41.670

Brian Ruf (EZD): open to the public, whether the only communication channels open to the public should be listed, or whether closed channels should also be listed.

54

00:09:41.940 --> 00:09:56.830

Brian Ruf (EZD): I think we're hearing that we should also allow the ability to list close channels. For example, the governments may offer, like Dod, may offer a communications channel that requires you to have, like a Gov. Or bill email address, one hundred and fifty.

55

00:09:56.890 --> 00:10:11.179

Brian Ruf (EZD): We would still want that listed, even though people outside of that community would not be able to join. But at least the Channel detail should indicate the join or linking instructions,

56

00:10:11.190 --> 00:10:13.920

Brian Ruf (EZD): but would include that requirement.

57

00:10:15.430 --> 00:10:20.210

Brian Ruf (EZD): So again, here are we missing other criteria. We should consider

58

00:10:20.340 --> 00:10:24.080

Brian Ruf (EZD): this the right list of details for the

59

00:10:24.340 --> 00:10:37.239

Brian Ruf (EZD): providing a communication channel on list, on the, on the, on the, on the Ons website. If you have, input drop it in the chat or send an email to Oscar at Moscow. I

60

00:10:42.830 --> 00:11:03.199

Brian Ruf (EZD): I'm. Going to turn it over at this point to Chris Roely. This is actually one of the highlights of the presentation Today is the Oscar registry, and Chris has dedicated a lot of blood, sweat, and tears to deleting this effort to try to get us to this point. The we actually had some

61

00:11:03.210 --> 00:11:21.460

Brian Ruf (EZD): Um, This is started out as an intern project. The the Grew. So it's been an interesting journey to get here. Um! And Chris has lived through it also. Ah! Without further ado, Chris, i'm going to turn it over to you and I'll cover the next two slides until you're ready to do the demo.

62

00:11:22.230 --> 00:11:26.709

Chris Roblee: Yeah. So uh good afternoon. Good evening. Good morning, everybody. Um!

63

00:11:26.850 --> 00:11:41.510

Chris Roblee: Before it is showing you with the latest, latest with the registry. So just as a reminder. This is still about Beta software. We're not in production yet. When we do have it completed it will be at this Url shared

64

00:11:41.540 --> 00:12:00.829

Chris Roblee: um. The purpose of the Oscar registry is to allow anyone to easily, anonymously share Moscow content and have a centralized repository. Take a lot of inspiration from like Docker, Hub and other repositories of the So of the lake

65

00:12:00.840 --> 00:12:02.110

Chris Roblee: out

66

00:12:02.320 --> 00:12:11.889

Chris Roblee: there's actually two ways of interfacing with the registry of both through a gooey which will be demoing as well as

75

00:13:40.670 --> 00:13:42.870

Chris Roblee: models of moscow

76

00:13:44.010 --> 00:13:56.029

Chris Roblee: so one thing that people found were that we had a lot of feedback is. They want a simple way to upload and convert and validate Moscow documents. So we

77

00:13:56.040 --> 00:14:04.429

have converters on the back end that will allow you to import in any of those three formats and export any of those three formats, so

78

00:14:04.930 --> 00:14:10.880

Chris Roblee: it useful as a conversion tool, validation and conversion tool

79

00:14:10.900 --> 00:14:14.499

Chris Roblee: we've built in basic search and filtering.

80

00:14:14.740 --> 00:14:19.610

Chris Roblee: I talked about validation, and so far we've been seating the

81

00:14:20.060 --> 00:14:38.080

Chris Roblee: reposit the registry with the this one hundred and fifty-three sixty-three metal pki, and is on baselines. But we encourage one once it's live the community to start uploading, and using the registry to host their content

82

00:14:39.410 --> 00:14:42.200

Chris Roblee: come to go in and jump into a

83

00:14:42.850 --> 00:14:46.630

Chris Roblee: really brief demo. Here, give me a second

84

00:14:47.900 --> 00:14:48.940

Chris Roblee: here.

85

00:14:53.360 --> 00:14:56.740

Chris Roblee: Ah, can you please release the screen?

86

00:15:01.650 --> 00:15:04.110

Brian Ruf (EZD): Ok. You should be able to take it Now

87

00:15:09.360 --> 00:15:10.540

Chris Roblee: he's.

88

00:15:10.550 --> 00:15:11.880

Can you see my screen?

89

00:15:12.160 --> 00:15:14.470

Brian Ruf (EZD): Yes.

90

00:15:14.480 --> 00:15:30.909

Chris Roblee: So really, simply, when we land here into the registry through the web App. This is an unauthenticated. User So this is publicly facing. You can easily browse search filter existing

91

00:15:31.820 --> 00:15:34.820

Chris Roblee: packages that have been uploaded by the community.

92

00:15:35.060 --> 00:15:45.680

Chris Roblee: As I mentioned earlier, we support catalogue component definition and profile models so easy to filter on those depending on what you're looking to do.

93

00:15:46.130 --> 00:15:51.779

Chris Roblee: I'm going to go ahead and search for a Let's see.

94

00:15:58.610 --> 00:16:15.129

Chris Roblee: Search for the Feder and Rev. For a low impact. Sas Basel in here. So immediately we have metadata here that's about high level information about the oscill content itself, and who uploaded it when,

95

00:16:15.280 --> 00:16:25.569

Chris Roblee: etc. We can easily view the document in three different formats. We discussed earlier

96

00:16:25.950 --> 00:16:34.019

Chris Roblee: and also download here. So this will just download it as a file in that that format here

97

00:16:34.070 --> 00:16:35.670

Chris Roblee: to drive.

98

00:16:35.980 --> 00:16:37.410

Chris Roblee: So

99

00:16:37.870 --> 00:16:39.980

Chris Roblee: um, i'm gonna go ahead. And uh

100

00:16:40.020 --> 00:16:57.549

Chris Roblee: this part is not working yet, I We'll get back to that in a minute. Well, one cool thing is that it does integrate directly with the oscill viewer, so we can go in, and we will share the like shared the link to our viewer which anybody can use today in the chat group.

101

00:16:57.620 --> 00:17:04.259

Chris Roblee: But we can go in drill down on any of the objects inside of that Moscow model.

102

00:17:05.030 --> 00:17:12.699

Chris Roblee: So ultimately we were thinking of integrating a lot of this into the same application. But for now it's still a different

103

00:17:12.730 --> 00:17:14.079

Chris Roblee: on it a different app,

104

00:17:14.190 --> 00:17:16.440

Chris Roblee: the viewer and the registry.

105

00:17:16.599 --> 00:17:18.440

Chris Roblee: So

106

00:17:21.150 --> 00:17:40.039

Brian Ruf (EZD): a couple of notes. Um. First of all, I see the great Gauss is is on, and the viewer was really, when when Ray was here, at easy dynamics, full time. The viewer is the product of a lot of his hard work. Um! So I just wanted to call that out. He's on

107

00:17:40.050 --> 00:17:48.160

Brian Ruf (EZD): the. As Chris mentioned, the viewer is available. Stand alone today as well as integrated. Just be a link

108

00:17:48.170 --> 00:18:06.220

Brian Ruf (EZD): to uh from the registry. Um. The important thing to know is that the viewer works entirely inside your own browser. So if you were to go to Viewer Scout that I owe whatever content you load upload into the view or never leans your browser never needs your computer.

109

00:18:09.520 --> 00:18:16.400

Brian Ruf (EZD): There was. It was designed that way with some semblance of security in mind.

110

00:18:17.230 --> 00:18:20.060

Brian Ruf (EZD): Yeah, that's it. Thank you. I'll turn it back to Chris.

111

00:18:20.630 --> 00:18:33.879

Chris Roblee: Great thanks, Ryan. Cool. So this is the unauthenticated workflow discussed next. I'm going to sign in on the back end Right now we're linking into.

112

00:18:34.800 --> 00:18:37.979

Chris Roblee: Let me go ahead and just use my gmail account

113

00:18:41.150 --> 00:18:50.690

Chris Roblee: so a little link with you with single sign-on providers through Gmail and I believe I know that ad i'm not sure which other ones Brian

114

00:18:50.700 --> 00:18:59.519

Brian Ruf (EZD): right now. It's just Google. It's It's a specially federated account we support Today we're open to federating other ones.

115

00:19:01.760 --> 00:19:11.619

So So anyway, here i'm signed in right Now let's go ahead. And so I mentioned earlier. We can

116

00:19:12.710 --> 00:19:20.679

Chris Roblee: structure and save information about this one. I've already bookmarked and liked.

117

00:19:22.530 --> 00:19:28.590

Chris Roblee: I can go back to my account. I can see which ones I've bookmarked here. So that was a

118

00:19:28.600 --> 00:19:37.300

Chris Roblee: highly requested feature some time ago, so i'm gonna go ahead and do something that upload a document,

119

00:19:39.060 --> 00:19:40.329

Chris Roblee: What? They

120

00:19:42.080 --> 00:19:44.700

Chris Roblee: go into my account here upload.

121

00:19:48.070 --> 00:19:51.049

Chris Roblee: I'm going to go ahead and upload the

122

00:19:53.570 --> 00:19:56.319

Chris Roblee: red for myired baseline profile.

123

00:19:59.600 --> 00:20:03.810

Chris Roblee: In fact, i'm gonna up with a few other documents, too, at the same time, just to show.

124

00:20:08.300 --> 00:20:22.099

Chris Roblee: I just selected eight different documents. So these are all very large, fairly large doctors. They're at least one megabytes size, some of the ten point five, ten, megabytes or so just currently the limit. So what's happened on the back end. It's not just uploading. It's also

125

00:20:22.110 --> 00:20:39.000

Chris Roblee: ah doing the full validation. So making sure a it's a valid Json or a yaml file, and then actually validating it against the schema itself um to guarantee that it it does meet the so. In this case there was an error. This particular file

126

00:20:39.010 --> 00:20:42.119

had one in there. That was I'd i'd modified

127

00:20:42.130 --> 00:20:46.019

Chris Roblee: anyway. So the other documents that successfully uploaded

128

00:20:54.200 --> 00:20:55.420

Chris Roblee: whoops,

129

00:20:55.430 --> 00:20:56.630

Chris Roblee: i'm sure

130

00:20:57.910 --> 00:20:59.750

Chris Roblee: let's give it a second.

131

00:21:12.190 --> 00:21:14.010

Chris Roblee: He's his life still better,

132

00:21:14.920 --> 00:21:17.680

Chris Roblee: I think I overwhelmed it by uploading too many documents.

133

00:21:18.040 --> 00:21:19.640

Chris Roblee: Right here we go.

134

00:21:20.630 --> 00:21:22.910

Brian Ruf (EZD): Ah, let me try here real quick

135

00:21:31.520 --> 00:21:45.350

Brian Ruf (EZD): Now I i'm getting four or four. Now, too, i'm not sure what happened that we crash the back end. This is why it's not been rolled out to production. Yet we're still like making sure these things, Aren't going to happen in production.

136

00:21:47.510 --> 00:21:53.539

Chris Roblee: Yeah, they give them a nice thing. I double check on the back end, but I think I overwhelmed it with A.

137

00:21:53.590 --> 00:21:54.899

Chris Roblee: It should be files

138

00:21:54.910 --> 00:21:56.970

Chris Roblee: that will go ahead for itself.

139

00:21:56.980 --> 00:22:03.629

Chris Roblee: But the idea here is you can upload any documents, large quantities at once, and then

140

00:22:04.380 --> 00:22:13.039

Chris Roblee: go in bookmarks on document. Manage delete them, if you like, and share them. So

141

00:22:13.810 --> 00:22:27.530

Chris Roblee: so basically that result is what it would. What we would have shown you earlier in the uploaded documents, Tab. So I definitely encourage everybody to once. It's out to start kicking the tire or start uploading, content

142

00:22:27.540 --> 00:22:36.239

Chris Roblee: and sharing public documents and using this as a launch as a landing ground for the

143

00:22:36.820 --> 00:22:55.990

Chris Roblee: sharing Oscar content. All right. So we're back up here. Yeah, I believe it was just delayed in the processing. Since we've done all that validation simultaneously, we're still working on optimizing performance. So here's all the documents that I uploaded here. This is associated with my profile.

144

00:22:56.170 --> 00:22:57.440

Chris Roblee: It's

145

00:22:58.350 --> 00:23:02.989

Chris Roblee: uh let's just click on one of these files. Um!

146

00:23:04.410 --> 00:23:06.350

Chris Roblee: I upload a year

147

00:23:06.470 --> 00:23:13.010

Chris Roblee: download again. Review the entire content in here, or, of course, look back over to the viewer.

148

00:23:13.460 --> 00:23:18.889

Chris Roblee: I go ahead and delete this if I no longer want it's shared in the in the

149

00:23:20.880 --> 00:23:21.930

Chris Roblee: the portal.

150

00:23:21.980 --> 00:23:41.659

Chris Roblee: So that's basically it. It's we're really hope It's very basic functionality. Right now. It does its job. We're working on scaling. So we can handle more and more load. We're also looking to add useful, basic, basic, useful features that people might find helpful to their jobs.

151

00:23:41.670 --> 00:23:50.280

Chris Roblee: So encourage everybody to when it gets up, and to start storing out their documents here, and please share feedback and requests

152

00:23:50.290 --> 00:23:51.220

Chris Roblee: he's.

153

00:23:54.540 --> 00:23:56.740

Chris Roblee: So i'll pause. There.

154

00:24:00.730 --> 00:24:04.289

Brian Ruf (EZD): Are there any questions uh we'd like to address right now.

155

00:24:04.300 --> 00:24:22.540

Brian Ruf (EZD): There's there's one in the chat that i'm trying to

respond to um by typing, but it might be easier verbally. The um. The the question is about what formats can be uploaded, and what does it do with the formats? Then all three

156

00:24:22.550 --> 00:24:40.719

Brian Ruf (EZD): formats can be uploaded. Xml. J son or Yaml. Upon receiving the upload it will convert to the other two, but that can take a few minutes, so if you immediately upload and try to immediately go over to the listing you won't

157

00:24:40.730 --> 00:24:47.160

Brian Ruf (EZD): you won't get all three formats right away, but within a few minutes they'll become available. It's an asynchronous operation,

158

00:24:51.670 --> 00:25:07.789

Brian Ruf (EZD): um and this is small. Files seem to be available almost instantly, but we we do a lot of testing with eight hundred and fifty, three, Rev. Five. It's the biggest file we've encountered so far. So that's that's what we're seeing there.

159

00:25:07.850 --> 00:25:17.169

Brian Ruf (EZD): I also want to point out that we do it. Intend to expand it to handle the other Oscar formats

160

00:25:17.420 --> 00:25:33.910

Brian Ruf (EZD): so that you know we we know you can't put a live Ssp. Up there, but maybe you might want to put an Ssb. Template or sample Ssp. Content. For example, we focused on the three formats that are most likely to be made public in the initial rollout.

161

00:25:37.950 --> 00:25:44.270

Brian Ruf (EZD): Chris, I see that it Michaela, is asking what it's using to process the conversion.

162

00:25:44.800 --> 00:26:03.100

Chris Roblee: Ah, yeah. So right now it's using the Jackson um Node Js library on the back end. So this is the node server. It's not the most performant, but it was chosen because it had the most consistent

163

00:26:03.130 --> 00:26:16.509

Chris Roblee: for my conversion rules, and it. So we are exploring

alternative modules. But for now, yeah, it's less performant than we'd like it to be.

164

00:26:17.200 --> 00:26:22.509

Brian Ruf (EZD): I think we're also using the um Saxon,

165

00:26:22.730 --> 00:26:24.590

Brian Ruf (EZD): H. E. Z. I'm. Sylvia:

166

00:26:24.600 --> 00:26:43.450

Brian Ruf (EZD): Yeah. Yeah. That's integrated with a typescript on the back end. Um. And so that's It's using the nist oscal converter exs Lt. Files in Saxon to be processed within. No, it is Chris just discussed.

167

00:26:46.600 --> 00:26:56.149

Brian Ruf (EZD): It's sex and live. That was, and I think I think that one would. You guys choose it for my time. But you chose that because it was the only one that could support that.

168

00:26:56.240 --> 00:27:09.929

Brian Ruf (EZD): Yeah, there's not so for those of you who aren't to where the Nist published converters require. Xsl. T. Three processing. There are not a lot of

169

00:27:10.350 --> 00:27:12.420

Brian Ruf (EZD): of choices from,

170

00:27:12.540 --> 00:27:31.750

Brian Ruf (EZD): you know. There's a lot of choices for processing. Xsl T. At one which is typically all you need on websites. But when you get into three Dio um sacks is one of the few that offers a a free opens. Well, it's not open source, but it's Free Library for doing that level of processing.

171

00:27:32.750 --> 00:27:39.249

Brian Ruf (EZD): I see email asks about the in the intended release date for the registry.

172

00:27:39.270 --> 00:27:43.819

Brian Ruf (EZD): Um! We would like to get it out during Q. Two of this

year,

173

00:27:43.870 --> 00:28:01.930

Brian Ruf (EZD): so we're near the end of Q. One. So you know, within the next, you know, two or three months is the goal. Um. We we. We have limited resources. We have a number of projects going on, and, as you see we are, we think our features are where we want them to be. At this point we have a couple of

174

00:28:01.940 --> 00:28:08.189

Brian Ruf (EZD): devops things to work out. We just want to make sure it's working smoothly before we make it available to the public.

175

00:28:11.430 --> 00:28:18.550

Brian Ruf (EZD): Oh, Wendell clarifies that the home edition of Saxon is Oss.

176

00:28:19.000 --> 00:28:22.019

Brian Ruf (EZD): Okay, intellectual properties and the paid versions

177

00:28:22.030 --> 00:28:24.199

Brian Ruf (EZD): so that that's helpful to now.

178

00:28:25.270 --> 00:28:35.710

Brian Ruf (EZD): Um, We We saw the Mikhail. We still a little bit more to cover. Once we're done with the registry, we have a couple more topics, and then we'll open the floor for questions and turn off reporting.

179

00:28:35.800 --> 00:28:41.830

Brian Ruf (EZD): Think I missed one? Tyler is asking, How can people? How can the community help the

180

00:28:42.310 --> 00:28:57.809

Brian Ruf (EZD): um? I think the best answer. There is certainly feedback. Um again. Oscar Oscar Io. Right now the registry repo is, we haven't made that publicly available yet. We've been on the fence about,

181

00:28:57.820 --> 00:29:15.769

Brian Ruf (EZD): and whether we were going to make the registry code

publicly available, the capability will remain publicly available.
Will The intention is for it to always be free. There will always be
an option to create an account for free and upload content for free
one hundred and fifty.

182

00:29:15.780 --> 00:29:24.649

Brian Ruf (EZD): The only thing that we may eventually get into
similar to like the Docker Hub model is

183

00:29:24.850 --> 00:29:36.380

Brian Ruf (EZD): is that we may do some verified um accounts at some
point in the future, and so there might be a nominal fee with verified
accounts just to cover processing costs.

184

00:29:36.530 --> 00:29:50.560

Brian Ruf (EZD): And, you know, handling costs kind of things to just
to again establish that. Yes, you're really getting this file from
this or from the Federal Pmo. Or you know, whomever is publishing it?
Not from somebody masquerading.

185

00:29:50.590 --> 00:29:51.810

Brian Ruf (EZD): Um,

186

00:29:52.400 --> 00:29:55.980

Brian Ruf (EZD): Chris, you have a question about filtering by license

187

00:29:56.020 --> 00:29:59.919

Brian Ruf (EZD): can one filter by license license?

188

00:29:59.940 --> 00:30:03.239

Brian Ruf (EZD): I'd like to be excited about that.

189

00:30:03.420 --> 00:30:13.539

Brian Ruf (EZD): So actually, I I guess, who asked that question? Are
you asking about the licensing of the content and the registry, or the
licensing of tools in on the tools list,

190

00:30:13.550 --> 00:30:14.800

Brian Ruf (EZD): but then

191

00:30:14.810 --> 00:30:16.900

Fen Labalme: content content in the registry,

192

00:30:18.610 --> 00:30:21.690

Brian Ruf (EZD): so that that's a good question.

193

00:30:22.330 --> 00:30:31.889

Brian Ruf (EZD): I believe we have. Chris. You do. We have a field free, but indicating the usage rights right now? I So the short.

194

00:30:31.900 --> 00:30:50.779

Brian Ruf (EZD): Okay. So that's we need to expand, to be able to um capture the usage rights and um, and then I guess we would offer the ability to to filter on that one of the features we were looking at actually is. The idea is, if you're publishing it in the registry.

195

00:30:50.790 --> 00:31:05.719

Brian Ruf (EZD): The actual content in the registry, Then you're You're making it available to the public for use. But you're right. I think we have to. There's a probably a legal we need to address in terms of making that clear.

196

00:31:05.730 --> 00:31:14.459

Brian Ruf (EZD): But in addition to that, we intend to expand the functionality so that people can list the fact that there's content

197

00:31:14.470 --> 00:31:30.139

Brian Ruf (EZD): without providing the content itself, and then point to a paywall. And so the use case we keep in mind. Here is the Iso, like If Iso wants to list twenty-seven thousand and one in the registry. Well, they want to charge you for it so they can list the fact

198

00:31:30.160 --> 00:31:47.899

Brian Ruf (EZD): that they have offscale content in the registry once they have it um, and then point to their paywall, where they would have any additional usage, terms and fees for getting to that content. We Don't support that today. That's one of our one of the next features on our roadmap

199

00:31:47.980 --> 00:31:49.610

to be able to support that.

200

00:31:55.340 --> 00:32:09.960

Brian Ruf (EZD): Yeah fence. I think you're you're clarifying that. Yeah, I think you're right. We do need to make this clear. Um for users of the registry that the well, first for people who are publishing to the registry that they are making this content available to the public one,

201

00:32:09.970 --> 00:32:18.080

Brian Ruf (EZD): and then for users of the registry that they can use it content freely, or if there's restrictions, they need to understand what those restrictions are.

202

00:32:24.240 --> 00:32:26.570

Brian Ruf (EZD): Maybe I'm: just taking a note on that.

203

00:32:26.670 --> 00:32:32.059

Brian Ruf (EZD): Any other registry questions before we move on in topics.

204

00:32:33.860 --> 00:32:45.669

Fen Labalme: Yeah, one more, please. Will there be mixed or in the content for the source of the content. So, like I can go back to the github,

205

00:32:46.150 --> 00:32:48.890

Fen Labalme: so I can make pull requests there if I want to,

206

00:32:48.900 --> 00:33:05.330

Brian Ruf (EZD): that that that'll go along with the um the feature to just point to content rather than upload it. Um! It. It would be the same feature at that point. So, in other words, you you would always point back to the source of the content

207

00:33:05.340 --> 00:33:16.789

Brian Ruf (EZD): in the in your registry listing when you uploaded it, or when you when you created the entry, and then maybe your choice is to also upload a copy locally for public use or not.

208

00:33:16.800 --> 00:33:23.420

Brian Ruf (EZD): Um, So what once we get to that, you know that that's

on the roadmap, but it's just it's not deployed today.

209

00:33:26.300 --> 00:33:28.190

Fen Labalme: Thank you. Very cool stuff.

210

00:33:32.850 --> 00:33:37.090

Brian Ruf (EZD): Yeah, We'd be getting a lot of excitement about this, so we're year to get it out.

211

00:33:39.120 --> 00:33:45.289

Brian Ruf (EZD): All right. Um, I am going to steal the screen. Share back here. Um,

212

00:33:45.680 --> 00:33:48.400

Brian Ruf (EZD): I see. Chris has already relinquished it,

213

00:33:53.160 --> 00:33:56.639

Brian Ruf (EZD): and continuing on here

214

00:34:00.220 --> 00:34:05.140

Brian Ruf (EZD): in January, when we presented this. Uh, when we presented the registry.

215

00:34:21.080 --> 00:34:23.460

Brian Ruf (EZD): I'm: sorry. Zoom crashed on me.

216

00:34:28.790 --> 00:34:30.889

Brian Ruf (EZD): You are still with us.

217

00:34:30.929 --> 00:34:33.449

Brian Ruf (EZD): Am I still sharing my screen?

218

00:34:34.050 --> 00:34:35.889

Michaela Iorga: Uh, no, The screen disappear.

219

00:34:35.900 --> 00:34:41.030

Brian Ruf (EZD): Okay, that was part of what happened with the crowd. Zoom Just seems to not like me sharing my screen.

220

00:34:41.040 --> 00:34:57.990

Brian Ruf (EZD): Um: Okay, So yeah, this is just some of the feedback we've already received, and it's factoring into our roadmap. Um again. If you have other things, you want to see, Oscar and Oscar that I owe um. We do want to hear what the community is interested in. That will affect our prioritization.

221

00:35:06.140 --> 00:35:30.179

Brian Ruf (EZD): Okay, I want to briefly touch on another capability that we're offering on that. This is more I I shouldn't use the term capability here. There's a specification. Um. The analogy my Cto likes to use is, you know, in the in the identity management world. Um, you know, there's Samuel and Samuel is both a data format and a data exchange protocol

222

00:35:30.190 --> 00:35:41.599

Brian Ruf (EZD): right now. Here in Oscar we Oscal is the format specification. But we don't have a corresponding data exchange specification.

223

00:35:42.150 --> 00:35:51.740

Brian Ruf (EZD): We believe that that's needed, and we believe that there should be an open source data formatic data, exchange specification for oscill.

224

00:35:51.750 --> 00:36:14.170

Brian Ruf (EZD): There's you know, the Fed R. And Pmo. Has talked about the need for one for for their purposes. Um! There is an issue out in the Nestosscow Github repo about the need for a rest. Api Um. So this is our take on one. We, you know we've designed it with use cases, the real world, Oscar, use cases in mind, and this is really intended.

225

00:36:14.180 --> 00:36:25.220

Brian Ruf (EZD): Ah! To be for exchange of oscill data um so or into org transfers, or, you know, transferring between one tool and another your oscill content.

226

00:36:25.290 --> 00:36:43.510

Brian Ruf (EZD): So we're releasing this draft specification as open source. The idea is that the final will be open source. So we are facilitating its creation. But we won't own its creation. The community will own it or the definition the community will own the

definition.

227

00:36:44.870 --> 00:36:46.029

Brian Ruf (EZD): Ah,

228

00:36:47.500 --> 00:36:59.199

Brian Ruf (EZD): we want to be able to handle. Have the rest Api handle of both. The off-scale formats themselves plus attachments. We want it to be able to handle any version of Moscow, any new models that come out down the road.

229

00:36:59.450 --> 00:37:13.430

Brian Ruf (EZD): Um! And we wanted to use traditional rest Principles um, for which there's no definitive standard out there. There's more like the industry. Best practices for rest.

230

00:37:13.440 --> 00:37:23.079

Brian Ruf (EZD): Um, and and recommendations. So that is a bit of a gray area. When we talk about rest principles that may facilitate or may generate some discussion.

231

00:37:27.520 --> 00:37:47.200

Brian Ruf (EZD): This is just an example. I I want to be mindful of time and allow additional questions at the end. So i'm going a little faster here. Um, this is. This is really out of laying the um, the methods and the endpoints that we're looking at. So ah, just in a nutshell. Ah,

232

00:37:47.210 --> 00:38:00.729

Brian Ruf (EZD): everything would be based on the the oscale syntax. So we talked about model name. It's the exact model name, as it appears in at the root of each of the oscale models,

233

00:38:00.740 --> 00:38:19.159

Brian Ruf (EZD): all lowercase, catalogue profile. What have you? So you know you can post a new catalog via the arrest. Um! Get catalog would give you a list of all the catalogs available in the repository. You would use the the get

234

00:38:19.170 --> 00:38:22.370

Brian Ruf (EZD): catalog slash identifier

235

00:38:22.550 --> 00:38:42.249

Brian Ruf (EZD): to talk, to, to deal with a specific catalog. So you can use that to get an entire catalog. Ah! To update it an existing catalog, or to remove one Um, there's this ability. Couple of ah endpoints for handling snapshots a couple of ah endpoints for handling attachments.

236

00:38:45.880 --> 00:39:03.859

Brian Ruf (EZD): The The idea is that um, you might say, get system security plan. You get a list of all the SspS in the system with their identifiers. Um, you would then use, You know you would find the Ssp. That you want. You would issue another get

237

00:39:03.870 --> 00:39:10.619

Brian Ruf (EZD): system security plan with the identifier of the Ssp. That you want to retrieve. Now you have retrieved it.

238

00:39:11.960 --> 00:39:15.779

Brian Ruf (EZD): Then you maybe want to attach the

239

00:39:15.790 --> 00:39:39.829

Brian Ruf (EZD): ah provide an attachment to that Ssb: so. Um, i'm sorry I've got ahead of myself here. Um, Now you want to. You want to upload a new Ssp. Into the system. So you. This is like, I'm. Delivering an Ssb. To another organization. So I would use posts System security plan to deliver my Ssp. And now we get an um an implementation assigned

240

00:39:39.840 --> 00:39:44.690

Brian Ruf (EZD): Ss. P. Identifier. How could they use that

241

00:39:44.750 --> 00:39:53.410

Brian Ruf (EZD): identifier to reference anything I want to do with that Ssp. In this case, I'm. Using that identifier to post a detachment.

242

00:39:53.670 --> 00:40:01.509

Brian Ruf (EZD): So I use another post command. I use the Ssp. Identifiers and say, i'm sending an attachment to this Ssp.

243

00:40:01.690 --> 00:40:19.129

Brian Ruf (EZD): Ah, I would then get back the Uu Id. For that attachment. The implementation would receive that attachment, and it would update the Ssps back matter content for the fact that there's now an attachment. Um. So we update the our link

244

00:40:19.400 --> 00:40:21.220

Brian Ruf (EZD): and um.

245

00:40:22.160 --> 00:40:25.789

Brian Ruf (EZD): You could then use something like, put it

246

00:40:26.480 --> 00:40:34.960

Brian Ruf (EZD): with the Ssp Identifier and the attachment identifier to update additional information about the attachment in the Ssp.

247

00:40:35.560 --> 00:40:49.839

Brian Ruf (EZD): So these will be. This is an example of how you might handle attachments within a rest specification. Um, These yeah urls are designed so that they can go inside the Href fields

248

00:40:49.850 --> 00:41:03.549

Brian Ruf (EZD): um within the Osc. Out content, and that it would work seamlessly. So, for example, if I have a a profile, the import Href statement could include the get

249

00:41:04.310 --> 00:41:07.419

Brian Ruf (EZD): uh endpoint for a catalog,

250

00:41:08.210 --> 00:41:15.329

Brian Ruf (EZD): it would just work across the Api to pull that catalog in without having to do any translation or modification.

251

00:41:17.330 --> 00:41:33.300

Brian Ruf (EZD): The The idea is that Oscar files work seamlessly with the rest. Api um, and i'm sorry as i'm talking because of my crash earlier. I've lost all my chat screens. So if anybody's posting questions to chat, i'm not seeing them. Maybe. Ah, Chris Orba Kayla can call them out.

252

00:41:33.410 --> 00:41:47.620

Chris Roblee: There's a question here about how the recipients verify

integrity of hospital documents. Exchange through the service. Um, John I. Measures are referring to through the registry or via this Api.

253

00:41:48.270 --> 00:41:49.410

Chris Roblee: We're both.

254

00:41:52.780 --> 00:42:09.699

Brian Ruf (EZD): I'm going to both. Both. Okay. Um. So the on the registry. One of the first things that happens when a file is uploaded, is It's put through the the Moscow valid, The the validation tools for whichever oscill model

255

00:42:09.710 --> 00:42:21.310

Brian Ruf (EZD): was uploaded. So you know the the X Xml. Ah schema that's published by Nest, or the Json scheme of this published by nest, and can be used for Json or yaml

256

00:42:21.480 --> 00:42:28.420

Brian Ruf (EZD): the file doesn't validate. Then it doesn't make it into the registered content. Doesn't make it into the registry,

257

00:42:28.430 --> 00:42:35.499

Brian Ruf (EZD): but but also there's an integrity right, the natural, like authenticity of the like, the creation who created it,

258

00:42:35.870 --> 00:42:52.489

Brian Ruf (EZD): the the authenticity right now is there. So for the registry There's no thank you for Clar by Chris. There's today people are asserting their own identity based on either. They create an account in the system with an email address, or they're using a

259

00:42:52.500 --> 00:43:09.100

Brian Ruf (EZD): a Google recognized. Ah account. Um! There's no additional validation today that's that verified publisher that we talked about introducing down the road. Um That would at least ensure the integrity of the source of the file.

260

00:43:09.280 --> 00:43:16.959

Brian Ruf (EZD): But the file itself. The whole reason that we do we display the file publicly,

261

00:43:17.460 --> 00:43:25.440

Brian Ruf (EZD): and in its raw format, and off in all three formats is so that it can be community verified.

262

00:43:25.450 --> 00:43:40.219

Brian Ruf (EZD): One of the other features that we intend to implement eventually are our community rating and ranking criteria, that you'll need an account to rate or rank the content. But we are looking at criteria, such as is it complete.

263

00:43:40.230 --> 00:43:59.060

Brian Ruf (EZD): Is it free of editorial issues? Does it? You know it? Does it do what it's representing that it does um. And so the you We want that to be a community-driven. Ah, verification, if you will, and you know we want people to give priority to the higher vetted content.

264

00:44:00.430 --> 00:44:11.350

Brian Ruf (EZD): Um, we recognize that one person might publish a component definition that's very sparse, and another person might then publish a better one that has additional detail.

265

00:44:11.360 --> 00:44:14.589

Brian Ruf (EZD): Um. And so we will want those both to get ranked accordingly.

266

00:44:16.240 --> 00:44:18.300

Brian Ruf (EZD): There's also no question kind of

267

00:44:18.310 --> 00:44:20.749

Brian Ruf (EZD): Okay, Chris. No. Go ahead. Chris.

268

00:44:20.870 --> 00:44:26.980

Chris Roblee: As a follow-up. Question about. Are we using open Api, or why don't we use on the opi, which

269

00:44:27.020 --> 00:44:28.849

Brian Ruf (EZD): we we have

270

00:44:28.860 --> 00:44:43.919

Brian Ruf (EZD): for the For the rest specification we are using open

Api um. So there's a yaml file, and we'll be making this deck available to the community. So you'll have these links. But

271

00:44:44.070 --> 00:44:49.870

Brian Ruf (EZD): first of all, if you, if you visit this this first link here you'll get to see the

272

00:44:50.060 --> 00:44:58.759

Brian Ruf (EZD): some additional write up about their specification, and then there'll be some additional links. You can view the open Api spec

273

00:44:58.770 --> 00:45:15.229

Brian Ruf (EZD): with this link here, and then the rest. The repository. For the rest, specification is a public repo. So you can actually go in and view the the issues you can submit issues you can contribute,

274

00:45:15.620 --> 00:45:18.889

Brian Ruf (EZD): and that's that's a dish repository right now,

275

00:45:20.880 --> 00:45:29.729

Brian Ruf (EZD): so we'll. I'll drop all these into chat at the end and again we'll make the whole deck available. Following this presentation

276

00:45:34.210 --> 00:45:37.509

Brian Ruf (EZD): did I answer the question? I feel like I may have drifted off the question.

277

00:45:39.780 --> 00:45:40.819

You?

278

00:45:41.890 --> 00:45:45.590

Brian Ruf (EZD): Ah! The question was about the open Api aspect which we are using.

279

00:45:45.600 --> 00:45:46.189

Chris Roblee: Yes,

280

00:45:46.200 --> 00:45:47.229

Brian Ruf (EZD): yes,

281

00:45:47.330 --> 00:45:52.349

Brian Ruf (EZD): okay. I think we've talked. I think we've covered the content verification that's on the roadmap.

282

00:45:53.510 --> 00:45:59.640

Brian Ruf (EZD): Yeah. So in the other Api spec the implementation will,

283

00:46:00.060 --> 00:46:11.599

Brian Ruf (EZD): you know there's There's detail that I don't have time to go into here about things like validated, you know, or you know, identity managed

284

00:46:11.610 --> 00:46:24.509

Brian Ruf (EZD): submission of content via the Api spec. So you know that's all going to depend on your level of identity, proofing to some degree the the implementation upon receiving content

285

00:46:24.520 --> 00:46:41.649

Brian Ruf (EZD): Ah should perform its own validation. The minimum that the spec requires is that it's syntactically valid. Um. Beyond that the implementation is free to do additional validation on the content as it sees fit for its particular purpose.

286

00:46:42.000 --> 00:46:58.719

Brian Ruf (EZD): So you know that it's always a tricky balance of where the spec should end, and you know, where should we should give the but but implementers freedom to move, and where the spec needs to be tight because otherwise tools won't interoperate well, and validation is one of those gray areas,

287

00:47:02.450 --> 00:47:05.149

Brian Ruf (EZD): any other questions on the rest. Api: So

288

00:47:05.290 --> 00:47:07.439

Brian Ruf (EZD): this is the classification.

289

00:47:09.290 --> 00:47:12.959

Brian Ruf (EZD): Our goal is to use the

290

00:47:13.060 --> 00:47:21.480

Brian Ruf (EZD): the appropriate portions of the rest api specification on the github. Sorry on the oscill registry.

291

00:47:22.340 --> 00:47:33.820

Brian Ruf (EZD): So the idea is that you'll ah your tools? Um! If they're configured to use the rest spec, then they're configured to interact with the registry,

292

00:47:35.050 --> 00:47:45.749

Brian Ruf (EZD): one of the use cases being the tool. If the tool comes across a ah need for a particular catalog that it doesn't have in its own library, it can query the registry

293

00:47:46.060 --> 00:47:50.689

Brian Ruf (EZD): to see what catalogs are available in the registry and offer them to the user

294

00:47:51.290 --> 00:47:53.149

Brian Ruf (EZD): by using the Api

295

00:47:53.510 --> 00:47:54.569

you

296

00:47:59.170 --> 00:48:08.880

Brian Ruf (EZD): okay onscale extensions I only have a couple minutes left, and I actually debated whether or not to leave this slide in, because

297

00:48:08.890 --> 00:48:22.200

Brian Ruf (EZD): since presenting it, we've learned that there's the possibility of implementing extensions, either at the Oscars specification level, or at the Meta-schema level, which is the

298

00:48:22.210 --> 00:48:37.569

Brian Ruf (EZD): the tool used to define the oscill standard um. So what we're looking for as we open the floor to discussions in a moment is, you know. Are you developing tools where you're using your own,

299

00:48:37.580 --> 00:48:56.180

Brian Ruf (EZD): your own Moscow? Ah! Extensions? Um! Do you have a need to use that namespace parameter on the on the properties. Um, do you have a need to an organizational need to define your own allowed values. And if so, um! Would you benefit from a standard

300

00:48:56.190 --> 00:49:05.059

Brian Ruf (EZD): where you can define all of those things such that tools can consume them and know how to validate your extensions to oscale content.

301

00:49:06.990 --> 00:49:17.989

Brian Ruf (EZD): So these are some of the questions to consider. Um, Michaela, this is a good point to open the floor for discussion, and i'm sorry we We went a little tight on time.