

Presenters:

Dennis Wuthrich, Farallon Geographics

Cyrus Hiatt, Farallon Geographics

Moderator:

David Myers, Getty Conservation Institute

So if you're expecting to hear about Arches version 7, you're in the right space. And once again, thanks for joining. Before we jump into the details, just a couple small housekeeping details here. Please keep yourself on mute for the conversation. But I do encourage you to ask questions as we go. So I think the way this could work is if you have a question, please use the Zoom tools to raise your hand or type your question into the chat box.

And Cyrus and I will make a point of stopping along the way to see if there are any questions. And if there are, we'll ask you to just pose your question if you raised your hand. Or if you've chosen to use the chat box, we will we'll read your question and try to take our best swing at answering. so that, I think, is how we'd like this to go.

I also know that we've got about an hour. And there's a lot to talk about. So I will just say that we're going to try to cover the full set of topics. And if it turns out that we need more detail, we can talk about organizing or scheduling a second call to go over the version 7 items in much more detail. Our topics today will really fall into these categories. We'll start with a quick description of where we are on the roadmap. And then we'll have a chat about a number of items that are very specific to Arches version 7. So let's dive in.

We have a roadmap for Arches. And if you'd like to see it and track how it changes through time, you can go to the archesproject.org website. There's a page specifically for the roadmap. What I'd like to do, actually, is point out two things.

The first is, there are really two versions of Arches-- Arches version 6 and, in fact, version 6.1 which was just released relatively recently, just in the last couple of weeks. And it provides a number of updates and bug fixes and enhancements. And it's worth knowing that we'll continue to support version 6 with bug fixes for at least the next year or so.

On the other hand, we're actively working on Arches version 7. And Arches version 7 is about to be released. It'll actually be released at the end of this month. And it has quite a number of new features in it. And you might be asking yourself, why have two versions of Arches? And the short answer is, not everybody needs to support multiple languages, and that's one of the key things in Arches version 7. So if it turns out that you're happy with version 6 and you don't need to worry about supporting multiple languages in Arches, you can continue to use 6 at least for the next year or so. On the other hand, if one

of your key requirements is to support your user community with languages beyond just English, then Arches version 7 is definitely the version for you.

I'm going to start with what I think most people see as the biggest aspect of Arches version 7. And that is this idea of internationalization, and what we mean by that. So in, I guess, the most succinct way possible, I'll say that for us, Arches version 7 includes internationalization. And what that means is Arches is now capable of supporting the display of its interface and the management of its data in multiple languages.

So you no longer are really constrained to a single language. And in fact, we've done a pretty complete rethink of how Arches supports multiple languages and what the implications are for managing data in multiple languages. And you'll find that there are many, many new tools to support multiple languages in Arches.

Some key items here is not only the ability to let you or your users decide what language to show the interface in, but also for those languages that whose scripts are read from right to left, there will be support in version 7 for that capability. So as an example, I believe Urdu is a language that is read from right to left, as opposed to English, which is read from left to right. And if you were to select Urdu as your language of choice, the user interface would reflect that.

And I'll just point out that internationalization includes more than just multiple languages and multiple scripts, but also the ability or the need to manage data and in particular, your string data in multiple languages. And Arches version 7 actually supports that. So a single node, a single string node can support values in multiple languages.

Then there's the concept of localization. And localization is really, I guess, in a way, a fancy way of talking about the actual translation. What it means-- to me, localization is the act of taking a text and translating it into a second text. And this is as much art as it is science. Hopefully, this little Twitter conversation gets to the heart of the matter.

Which is to say when you build an Arches application and you build it in a language, someone is going to need to translate that into a second language. And there's some effort and some-- I suggest some art into that.

So in a nutshell, when we talk about multiple language support in Arches, we're talking about internationalization, the ability to support multiple languages and the data that you need to collect in multiple languages. And that's something that we've built into Arches. The actual localization, the translation part, is something that really is going to fall to the user community.

And we actually have some tools to help you support that. We've already set up a site called transifex that allows community members to take the Arches, the core Arches application, and translates all the static strings in Arches into alternate languages. And here's the little screenshot that shows a bit of the status.

The core language for Arches is English. And there are tools in place to support the translation of all the core Arches strings, the static strings, into multiple languages. And this is just one of many tools that you may use to support the translation.

And they're helpful. That is really, for sure, the place to start. But I want to point out that-- and I've sort of blithely used this terminology of static strings. The static strings are the strings that are part of the Arches core user interface. But of course, really the key aspects of Arches is that allows you to define your own models and your own data entry forms.

And these are not static strings. These are what we call dynamic strings. And these cards, or data entry cards, for example, also will need to be translated into the languages you want to support. And since you build your models yourselves, Arches comes with the tools that you'll need that you'll want to use to support creating the labels in your data entry cards in multiple languages.

And that's a very quick summary of what we mean by internationalization and localization in Arches version 7. And I think it's maybe a good place to stop and see if we can answer some questions. And David, I believe, you're tracking the questions. Maybe you could see if you could read the first couple of questions.

Dennis, I'm watching for questions. I don't see any yet.

OK. I think there might be some in chat. No?

I've seen comments, but no questions yet.

OK. Perfect. Well, then thanks for checking. Cyrus, is there anything else around internationalization that you want to add before we move on to the next topic?

No, I'll just point out that-- yeah, so you have your localization in core Arches. And Arches will ship with languages or translations for core. But then you'll have this separate task of translating your graphs and your card labels and names, as you mentioned. So there's kind of a division there. But yeah, I think that was it.

Great. Thanks. So maybe just to hammer that point home. You're looking at a screenshot here of a model from historic England. And when I say that, there will be some work for you to translate. What I mean specifically is in this example, it will be up to you to decide how to translate designation and protection assignment, and designation name, and grade, for example, into the languages beyond English that you wish to support.

And the reason for that is these are your models. And you've designed them completely independently. So although there is some work for you where you won't have to worry is where it says Card Designer, or where it says Card Configuration, or Settings, all the things that come as default by Arches. Those strings will already be translated for you.

Dennis, in case you didn't see it we did have our first question through chat from Karl Fogel asking what languages are supported already. Are there any?

Thanks for the question, Karl. There are a number of languages that have already gone through community translation for core arches and, I think, have achieved 100% review. And off the top of my head, I'll say those languages are French, Urdu, and Bulgarian, and, I believe, Hebrew are the languages that I know for sure have been fully translated.

I also know that Arabic is in flight. I believe Russian is in flight or is potentially completed. And I believe Spanish is also in flight. And of course, any other language that you, as a community member, feel like you'd like to support. There's absolutely nothing stopping you from tapping into the community and asking for help in creating a set of translations for really any language that you find interesting. Are there any other questions, David? You're on mute.

Yeah. Just a comment from Crystal saying that Arabic is up to 30% now and getting better.

Excellent. Let's move on. And our next topic is user authentication. And really, the key thing here is in version 7, we'll be introducing what's known as multi-factor authentication. And in our case two-factor authentication. And I'm going to ask Cyrus to give a quick summary of what we mean by this.

Yeah. So of course, currently in Arches, user just authenticates with a password. But in version 7, admins will be able to enable two-factor authentication. So users will then be able to use an authenticator app to generate a token to verify their authentication. And that is implemented-- or at least an admin has two ways that they can implement that.

They can require it and force all users to use two-factor authentication. Or they can make it an option. So the user has the choice to use two-factor authentication or not. And the mobile applications that people can use or any popular two-factor authentication app like the Google Authenticator, and there's an open source one called [INAUDIBLE] Or an admin can choose not to use two-factor authentication at all if that's what they want.

Thanks, Cyrus. Any questions related to that?

There's a new question from [? Mahmoud, ?] asking, is FIDO2 supported physical [INAUDIBLE]

I'm not familiar with FIDO.

So it's like—

[INTERPOSING VOICES]

I'm not sure. I'm not familiar with that, either. I'm sorry, [? Mahmoud. ?]

That could be something we check into.

Yeah, yeah. We'll look into that and get back to you.

Any other questions, David?

Not yet.

Let's move on to our next major topic and that is the import and export of data into Arches. And I'll just say that if you've been in the position of having to import large data sets into Arches, you know that it's a pretty big lift. There's a lot that goes into data import and data export.

And one of the main things that we're aware of in Arches is that there are some restrictions or constraints in how you structure your data for import. And we're aware of that. And in version 7, we've relaxed a lot of those constraints. And I'll show you an example of that in a moment.

We also know that loading data into arches can take some time. And the reason for that is because we spend a lot of time doing data validation when we ingest data, and that data validation can be quite expensive. And then once the data are loaded into arches, then we index the data into Elasticsearch, which also can be quite expensive.

So part of what we've done in version 7 is to try to simplify the process and to try to speed up the process. And I'll show you a couple of ways that that's possible. The first is, really, this introduction of what we're calling the ETL manager, so apologies for a really unsexy name. We probably couldn't come up with anything worse than that. But there you go. That's what we're calling it right now. Definitely, welcome to suggestions for improving the terminology here.

But what we mean by this is, as you can see, an interface that lets you import and export data. And the idea here is it's a way of including a whole data processing module into the Arches platform. And it's designed to let a developer create an import, essentially an importer and an exporter. And we've built an importer for a single CSV file.

So now instead of having to put your data into an Arches-structured CSV file, you can take any CSV file that you'd like and use the importer here to identify the model that you want to import into and then map the columns of your CSV file into the nodes of the target graph. And Arches will do a quick validation and then start the data load.

And that data load will be done asynchronously, which means that once you start the load, you can move on to other parts of the Arches application. You don't have to wait for it to finish. And arches will then keep track of the state of your import job. And it will let you know when it's done.

I can tell you that our initial testing has shown that this approach is quite a bit faster than our current version 6 data import tools. My own testing suggests that taking a file that might have required an hour's worth of processing using Arches command line tools can be processed in just a few minutes with this importer.

So there is a significant performance improvement here. And it's just a lot easier because there's a user interface. You don't have to have command line access to charges to do data import.

I guess the other thing to point out here is this module has specifically been designed to be very flexible and allow a developer to introduce an importer for really any format that you find interesting or worth working with. So if you're working with structured import, structured files from third parties, you could write an importer specifically to support a very specific format. And the reverse is true as well.

You can write an exporter to export Arches data into specific formats if you wish to. And maybe that's a good place to stop and see if there are any questions.

Can I add one more thing, Dennis?

Oh, please do.

This also gives you the option to undo a load. So let's say you loaded some data and you decided, oh, there was something wrong with it. You could actually reverse that load, make your changes, and then rerun the load again. So it's also a nice feature.

For those who need that, yeah.

For those who—

David, are we good with questions?

No questions. But there's lots of nice and very much agreed this is cool-- so some feedback.

Excellent. Well, thanks for the feedback. I'll just tip our hand a little bit here and say that right now, we've got the setup for importing data and exporting data from Arches. I think we were already talking about how to extend this ETL manager to include scripts for bulk updates of data already in Arches.

So imagine the case where you've uploaded, say, 100,000 new instances, and you realize that you misspelled a common word. And instead of having to unload and reload, we think we'll be able to provide, say, find-and-replace tools for making that kind of bulk update easier and possible.

We do have one question now from Marcus, which is, when you write an importer, it means you can reuse that mapping over and over in the future? That's a question.

Yeah. Actually, Cyrus was one of the main designers of this model. So Cyrus, maybe you can provide a bit more detail there.

Yeah, so if you wrote your own custom importer, then yeah, you could reuse that mapping over and over. There's endless possibilities really of what you can design for your importer, although designing an importer does require some Python knowledge and JavaScript as well. These are components, like other components in Arches.

One nice thing to point out here also, that's because it's a really good question, is as the community writes additional importers and exporters, those should be shareable. So you only have to write that importer once or that exporter once.

And then others who might find it useful can take advantage of your good work in the same way that you can take advantage of the work of someone who's put the effort into writing an importer that you find useful. So this is a good place, I think, for the community to see value in supporting one another with essentially shared effort.

One new comment from Thomas Hewitt that bulk update is the next big issue.

Yep. Bulk update is definitely on our radar. We have also introduced another way of working with data in bulk in Arches. And really, they boil down to what I'll just call database tools. So we've introduced this ability to work with Arches directly at the database level.

So if you know SQL and you're comfortable with using SQL to manipulate data, we have introduced a series of essentially helper functions that you can access directly from Arches' transactional database, which happens to be Postgres. And you can use these helper functions to create, update, delete, modify as you see fit data directly inside the database.

And just to give you a sense of what's possible here, as you write your models, you can use a helper function to represent those models in traditional relational series of relational views here. One thing to remember in Postgres, views are updatable. So what we've done here is provide you with what amounts to a relational database of Arches models, which allows you to use just straight out of the box SQL tools to update tables.

So you can insert data into a table. And in doing so, Arches, behind the scenes, will automatically sync your changes and in this relational schema to the transactional data in Arches core graph-based schema. This is a case of "with great power comes great responsibility" kinds of things.

It's possible to do a lot of really, really sophisticated data manipulation here. And you should do that if you know what you're doing. This is the case also where a wrong turn puts the onus on you to fix it. Now I will say that one of the things we've implemented here is managing the transactions with specific unique identifiers.

So any transaction that you implement in Arches, whether it's versus the user interface or here at the database level has a transaction ID associated with it. So you can always use that to roll back things that you think might have done in error.

But really, the point I want to make here is for those of you who are technically inclined and really feel like you've got confidence in your SQL chops, you've got a brand new and really sophisticated and very high-performance tool for working with Arches data. And again, a good place to stop and see if there are questions.

A comment from Ash, that this ETL manager looks great the pluggable. Importers with the common UI is very well-engineered solution to a hard programming problem. Really impressed.

Thank you, Ash. You can thank Cyrus. He did all the lead development there.

Thanks, Ash.

Last thing I want to say about data import and export is that we've spent some time really thinking through enhancing the way in which Arches integrates with Esri GIS ecosystem, both at the Esri server level, but interestingly, and maybe most importantly, at the Esri desktop level.

So Arches now supports pretty high-performance, real-time, or essentially near real-time integration with ArcGIS Pro. And what does that mean in plain English? It means that you can configure ArcGIS to present its quite sophisticated instance data as if they were feature layers in ArcGIS Pro, and that you can use ArcGIS Pro to create new instances of models or update existing instances of models directly inside the Esri application.

And it requires the use of an add-in that we've written for Esri, freely available. All you have to do is download it, double-click on it. And it'll install itself in your ArcGIS Pro, and then configure Arches to default-- you can configure Arches to decide how you want to present your instance data to ArcGIS users. And off you go. You can start using the full power of both Arches and Esri in one environment.

I'm working on putting together a little video that demonstrates this. But the key point here is we've got it working. And I want to give a shout-out to Historic England, who've done some really good work in supporting this. So thanks to Andy Jones and his crew at Historic England for helping out with this. I guess the last thing I want to say here before seeing if there are any questions is to point out that the way we're doing this-- and maybe it's-- I'll step back a second and say it's interesting that one of the first things we've integrated with is a proprietary GI system. So Esri, as you know, is proprietary.

And Arches is open source. That doesn't preclude the ability to provide an integration between these two technologies. And we've done so in a way where we've leveraged open standards-based ways of

presenting the data. So the good news here is both Arches and Esri can speak the same open standards format of geo data. And we've used that to really foster this connection between Arches and Esri.

So I know an image here or a little video would actually be worth all the words I've just spoken. And I will be working on that. But hopefully, some of that made sense. And I'll see if there are questions.

Dennis, there are two questions from Junaid. The first-- I'll put the first one, which relates to the Esri integration, is QGIS also under consideration for integration?

QGIS is definitely under consideration for integration. And honestly, probably, it'd be a lot easier to integrate with Esri. So it's definitely something that we think is a really good idea. And I think it would be great. We don't have any funding to do that yet. But if there's interest out there amongst the community to take this on, I, for one, think it's a great idea.

And Junaid had another question relating back to data import which is, can data be divided in insert and append operations between new CSV importer and relational schema?

Well, I'll take the first swing at that. There's nothing that would keep you from using both the ETL module and the relational schema in ways that make sense to you, even if that means divvying up how you choose to modify your data. Cyrus, do you have anything to add to that?

No. I can imagine you might want to do the lion's share of your load using the relational schema where you'll really be able to leverage the performance and then maybe users can then append data, additional tiles to resource instances. That seems like a viable pattern for sure.

Other questions, David?

Not at this point. No.

OK. Let's move on. We've also spent a lot of time thinking about just overall performance improvements, not just performance improvements with data import and export, but also performance improvements at that kind of at the full application level. And there are kind of two things that I wanted to highlight in today's conversation.

One is this idea of a published model. And what does that mean? Well, right now in Arches when you-- anytime you ask Arches to show you information about a model-- so when you're working with the editor, or when you're doing certain search results, or when you're asking for a report, Arches ends up having to build the structure of the model dynamically every time you request it.

And it does that because one of the key strengths of Arches is it lets you build these models dynamically and interactively, and you can change them over time. But there's a cost for that. And the cost is to having to rebuild the current shape of the model every time you need to know about it.

So what we've done in version 7 is introduce this ability to essentially what we call publish the model. And that means when you've got the model in a shape that you like and you're unlikely to change it any longer, you can publish it.

And the publishing means essentially saving a serialized version of the model so that instead of Arches having to rebuild it dynamically every time, it can just go straight to the saved essentially version of your database of your model and use that to enter as a way of building it entry forms and responding to search requests and report requests and that sort of thing.

And there's a lot of technical detail here, which, if there are questions, I'm happy to dive into. I think, for this audience, the thing to really note is that publishing models is a performance enhancement. And as a really nice byproduct, we end up keeping a copy of how your models have changed through time.

So if you make a model change, we keep track of every version of that model in its evolution. So it would be possible to reuse earlier versions of models if you wanted to.

And then the second idea here is, what I've just called webpack because that's what it's called. Webpack is a way of essentially minifying so building the web interface and making the amount of information that needs to get sent from the Arches server to your computer smaller. So that should increase the performance. It should make arches seem faster.

And that's part of version 7. It has-- in addition to some very immediate performance enhancements, it also provides some additional things that, we think, are pretty interesting one of which is the ability to support the use of alternate front-end libraries for creation of Arches user interfaces in the future.

And this is a big deal because some developers really, really like particular UI libraries. And we've done some early testing with angular-- I believe, is it, Cyrus?

React, actually.

Oh, sorry, React, where we can now write or rewrite existing Arches UI or create new UI using React, for example. But essentially, that would be true for any front-end library that you like. And I guess I'll stop there and see, Cyrus, if you wanted to add anything to either of those-- either the published model or webpack stuff.

I think that was a really good summary. Yeah. Improved performance in two ways will reduce the demand on the database or the number of queries with the published model and be able to compress files to deliver to the front end much faster. So yeah, sounded real good.

Thanks. Any questions about this, David?

No new questions.

Excellent. We've done a lot of work to ensure that Arches maintains consistency with some of its key underlying technologies. And you get a sense here of what some of them are. But things I wanted to highlight in particular are we've upgraded Arches version 7 to use the latest long-term release of Django. And that brings with it some performance enhancements and some pretty significant bug fixes as well. So the next version of Arches will take advantage of the improvements in Django, as well as Postgres, where we're running Arches version 7 on top of Postgres v14, which is the latest production version of Postgres. It remains awesome. And we also can take advantage of Postgres version 3, which is a kind of key extension to Postgres and provides a lot of geospatial functionality, also remains awesome and fast. So we get to use the latest versions of both those key technologies.

We're also updating Arches' search engine to Elastic version 8. It provides a host of additional new capabilities. And one thing that immediately caught our eye is Elastic claims a performance improvement of 20% on indexing data. So that should also be something noticeable for our Arches community. Within Farallon, we've switched over to doing our Arches development and deployment using Docker. And that's turning out to be really, really nice, very efficient and effective way of standing up Arches and managing its dependencies and is the kind of thing that would be really-- I guess we would say-- a really, really good thing to look at if you thinking about production deployment.

And then finally, we use a number of libraries in Arches for all sorts of helper functions. And many of those libraries have gone through their own improvements and evolution. So we've updated some of the key libraries that Arches uses, in many cases, in its user interface. So we've done technology updates throughout the application. And I think we're pretty happy with where things stand there. Cyrus, anything to add?

No. Yeah, other than-- let's see. If you're doing development or you're working with 6 and you're considering moving to 7 and you need to run two versions of Elastic at the same time for your different implementations, then Docker has been really helpful in allowing us to develop against 6 and 7 simultaneously.

We have a new question from Karl, who says, Elasticsearch's recent releases are technically great but are under a not quite open source license anymore. Are alternate search libraries on the roadmap? Yeah. We are aware of that. And we have been talking about alternate search libraries. And there aren't-- I know they do exist. There's lots of reasons why moving to them is either problematic or a lot of work. I will say one thing we've been looking at is OpenSearch. And we made a conscious decision to not replace Elastic with OpenSearch, mostly because it would require that you run Arches on AWS, which we don't want to do. So Cyrus, do you have anything to add there?

No. Other than we're just keeping an eye on the situation. And maybe if OpenSearch becomes more popular and is more available, there aren't the limitations that exist now, then-- we'll just keep our eyes on that. And we would like to support open source or ensure that our dependencies are all open source possible. So yeah, thanks for the question.

Any other questions, David?

No.

All right. Which brings me really kind of to the end of this very, very quick tour. And that is to say that we're working on Arches version 7 right now. And it is actually publicly available. It's on the Arches repo-- there's a version 7 branch-- which means that you can test it. You're able to hold the code and start looking at it and deciding what you think of it. So you can test Arches now.

And by the way, you should test Arches now because it's a major upgrade. And there are a lot of new things there. And my pitch to you is the sooner your technical folks put some eyes on the new version of Arches, the sooner you'll start to get a sense of what's possible, and also the better it is for the community.

So the more people working on testing and identifying bugs and working with the codebase, the better it is for everyone. So if you do have the flexibility of taking a look at Arches version 7 and you've got some technical folks that would like to dive in, consider this an invitation. We would love it for the community to start putting Arches version 7 through its paces.

We're under no illusions that it's perfect yet because it isn't, and all the more reason for more people to look at it and identify places where it needs attention. But it is available. It does work. And it would be lovely to have feedback from the broader community. And I think I'm going to end with just reminding you that we have a release date set. It's August 31 of this year. So really, at the end of the month, that's coming up pretty quickly.

We're busily at work at Farallon on finalizing the first release of version 7. If you've got some time and you want to help, please do so. But this is a good place to stop and see if there are any last questions. We do have one new question from Ash. Technical question, the process of upgrading from 6 to 7, how much work will it involve? Presumably, it's not a straightforward git pull restart Apache like most minor upgrades are.

Thank you for that question, Ash. This really was the question that I was expecting someone to ask and is a major reason why Cyrus is here.

Yeah. This is going to be a bit more work than a typical upgrade for, say, a feature release. There are some changes that will be required to all of your custom templates, not huge changes, but some modifications. We'll also, of course, have to update your dependencies like Elasticsearch.

And then there are some other changes that you'll have to make potentially on your Apache server. But it's well documented already. We'll come up-- present the steps to upgrade as we have before. It won't be as bad as or as difficult, I should say, as upgrading from say 3 to 4 because there aren't any major

changes to the data model or anything like that. So certainly, a little bit more work than a typical feature release, but that's certainly doable.

Can I just add one or two additional thoughts? One is, because internationalization requires us to manage string data in multiple languages, one thing to note is as you start thinking about importing strings into Arches in version 7, at least early on, you'll have to think about how you've configured your Arches instance, what languages you've configured.

And there's some onus on you to make sure that you're consistent with the data that you import and the languages that you've configured for Arches to support. So there are-- it's not really an upgrade, it's a result of after upgrading to Arches. You'll want to think a little bit more carefully about how you representing your strings and data versus the UI. So there will be some additional things to worry about from a systems administration standpoint, too.

There is a new question from Jim McGowan. This may not be the right venue for this question, but, are there any updates on the status of a future collector app?

Good question, Jim, and thanks for posing it. Well, here's what I can tell you. This upgrade to version 7 and the inclusion of internationalization has, for sure, broken arches collector, so it won't work in version 7 for a number of reasons. So we know that. We're in the midst of thinking about how to support a next version of Arches collector.

And we're doing the initial engineering, research, right now on that. We've got some, I think-- actually, amount to some pretty interesting ideas about how to essentially rewrite Arches collector for version 7. And that gives us an opportunity to recalibrate not only the approach but the technologies we're using. So I am thinking about that now. And I know that the Getty has asked us to start putting together some thoughts on overall effort and both time and costs for that. So it's definitely on the radar. Nothing has been approved or authorized yet. But we are absolutely thinking about that. So apologies if that only partially answers your question. But that is really that's where the state of collector stands right now. If there aren't any other questions, I'll just say thanks, everyone, for attending today. Hopefully, you found this useful. I know it was a pretty quick sprint through all the new stuff that's coming and kind of purposely at a high level. If any of the things you heard about today are of deeper interest to you, let us know, and we can talk about deciding whether it makes sense to have another group call or if there's interest in exploring how to test arches.

And if you need a little guidance, let us know. We're happy to get you pointed in the right direction. But really, my bottom line is I really appreciate you hanging out with us today and hearing about what we're doing. Thank you so much for that. And I guess I'll stop sharing my screen here and ask one more time if there are questions that anyone has. We would love to ask before we sign off. No? Well, again, thanks, everyone. Really appreciate it. And best of luck with your Arches work.