

**Presentations:**

- **Warden: A new home for the Ancient Monument Index for England**  
(Andrew Jones, Developer Team Leader, Historic England)
- **First Steps of Arches on Kubernetes**  
(Sarah Byrne, Product Manager, Flax&Teal and Phil Weir, Director, Flax&Teal)
- **Heritage Monitoring: Florida Public Archaeology Network**  
(Adam Cox, Owner, Legion GIS LLC)

Well, good morning, good afternoon, and good evening, everyone. Thank you so much for joining today's Arches Developer Meeting. I'm really super stoked that so many of you decided to participate today. It looks like we've already got a great list of participants. So thanks in advance for joining us. I really appreciate that.

My name is Dennis Wuthrich, and I'm the CEO of Farallon Geographics. Many of you probably recognize the name Farallon. We've been working on the Arches project for quite a number of years now. And we had an internal conversation, back early part of this year, back in, I guess January, February. And one of the things we thought that will be really great would be to start a conversation amongst developers in the Arches community.

And by way of jogging your memory, we had a conversation back in August where we had our initial Arches meeting. And although we wanted that to be done in person, events conspired against us. And instead we had an online meeting, which I thought was really, really good. And it was great to get to meet people, hear what people are working on, put names and faces together. So I appreciated that.

And at the tail end of that meeting back in August, we took a StrawPoll, and it was pretty clear that people were interested in having a second meeting. And that the second meeting should be a chance for members of the artist community to share their work. And that's what we're going to do today. We're going to have three members of the community talk to us about the work that they're doing with Arches. And I'm super excited about that.

Our format for today is going to be, a bit less interactive and more presentational, primarily because we've got three different teams who want to show their work. And it's just a bit unwieldy to have 30 odd folks chiming in an ad hoc way. So we thought it would make sense to have this be more of a webinar presentation, at least initially. And the way this will work is we'll have three presentations. They'll each have about a half an hour to present their work, at the end of which we'll do a quick Q&A session. And I'd like to encourage people to type in their questions into the Zoom tool. And that would give me a chance to pluck out the questions and pose them to the speaker. And then we'll launch into the next meeting.

At the end of the third presentation, if we've got a bit of time left and there's interest, I think it makes sense to open up to general dialogue then. So that's how we'll be organized for the next hour and a half or so, two hours or so. Despite the fact that everyone's on mute, please don't let that dissuade you from chatting amongst one another. Like I said, posing questions. That would be really, really great.

Today's time is really all about seeing the work that our colleagues are doing. And I'd like to start by having Andy Jones who is with Historic England, start off with a presentation that he's working on for Historic England. And Andy, I bet you're ready to go. I'm going to hand off the conversation to you. And I'm going to stop sharing my screen. And I think that would put you in a position to actually start sharing your screen and starting your conversation. So without further ado, Andy Jones why don't you take it off.

Thanks very much Dennis. OK. So I've got to see how I can get this to work. So I can't present at the moment.

Let me see if I can make you a presenter.

OK.

OK. You are a host now. So you should be able to present. ,

There we go. Let's minimize that down. OK. Thanks, everyone. Let me just get this presentation on the go. Can everyone still see that?

Yes.

Great. OK. So thank you. Yes, I'm just going to spend hopefully the next half hour. If I can't keep it within half hour, please tell me if I'm going to go over. Just to give a brief intro to the Warden, which is a project we've been working on for 18 months or so now. And it was a project to replace an internal system of Historic England.

So just very briefly on what we do. We are a government arm's length body in England. We work for the Department of Culture, Media, and Sport. And our role as an organization is to really champion historic places. And raising the profile and the importance of heritage in England. Identifying, protecting our heritage, including things like recommending designations of sites for protection. And also doing more work in understanding historic places. We also support change such as providing help and advice to the government on policy, and also expertise at a local level within our local planning authorities.

So again, swiftly moving on. One of the tools that we've had over the years that's helped us on this task is a system-- it's called Ancient Monuments Index for England. But we know it internally as AMIE. And this is a very rich record of the English historic environment. And has been a system that we've slowly accumulated over the years. It started off with simple trust remodeled decades ago. And has slowly been bolted onto with information about marine

maritime, sites, crashed aircraft, as well as activities that are occurring around to do with investigation and research.

We use this tool to not only collect that research and investigation work, but we also have used it for providing information. This information is provided to members of the public, commercial organizations, academics, as well as for the wider government applications such as national infrastructure projects for instance. And the system itself has been collated-- well, over 20 years. The system itself is more than 20 years old, it has close to if not over a million records. And consisted of around 400 tables within the database. So actually taking this system-- migrating it was quite a huge undertaking for us. And it's probably the reason why it took so long to get around to--

So this was going to be the new home, Warden. I can't remember what it actually stands for, because it's been a little bit of an in-joke. But it's a keeper of the data now. I've put that little asterisk to say that it's full now. We have an ongoing program now, where we are starting to hand off this national data set to the local historical federal records around England, who will become the owners of that data. And the system will move more towards a research tool that will allow us to collect data and send that off to the respective local authorities.

The system itself, we've delivered the second version of it already. The first version we started in 4.4.1, last year. And that was released earlier this year in the summer time. And since then we've released the second version at 5. So we spent the last six months or so upgrading that and adding additional functionality. As with all great Arches applications, it all starts with models. So Phil Colaly and our data team spent quite a bit of time working through the intricacies of those 400 tables, and developing for us seven, it's actually eight, but seven target models that we've populated with that data.

So I just wanted to really start off saying, data wrangling as we came to call it, was quite a monumental process. In the end we managed to migrate across and generate around 800,000 resource instances. And this is a little flavor of what Shantelle spent many, many weeks building up, in order to allow us to extract that data, that detailed process of extracting the data, doing cleansing, checking the data, putting it in a way for the business to also check it, and then also to transform it into CSV or JSON, if the data structure is slightly more complicated with a bit more hierarchy. So really didn't want to glaze over that and go on to the actual application itself without talking about that.

So apologies for the diagram, it is a little bit of an eyesore there, but really this is a very high level component diagram of what we've been working to, and what Warden currently sits at. And I've highlighted five main areas that we have been developing and integrating with over the last couple of years. And I'm just going to swiftly move on, rest everyone's eyeballs on a Friday evening or morning, and move to this.

So this again, it just shows the main areas that we concentrated our work on. Obviously in the middle is the Warden application itself where the data work was happening, the Arches extension work around those types of functions, those standard ways of extending Arches. Also

along the start of the bottom, we had a number of requirements around being able to do import-export of that data. So importing in order to allow us to update those records en masse as the accession in work progresses over the next few years.

And also as part of the advisory work that we do, we need to be able to generate very large volumes of reports for organizations. And sometimes like we said, they can be national projects. We are in Esri house, so we have quite a few workflows within the Esri ArcGIS stack. So we had to make sure that the Warden-- sorry, the data that was in Warden was able to be accessible by that. We could also do some querying between them, and also allow ArcGIS to be able to help us digitize some of the features that we're going to be added into Warden.

And then along the top, the Heritage Gateway. Warden is an internal system, we don't publish the user interface, but we do allow access to a subset of the data through a site called the Heritage Gateway that we provide a service for now. And then the last one was, we needed to be able to integrate with our mapping services, that host ArcGIS online. And also to be able to have system monitoring through application in sites in Azure.

Again swiftly moving on. The actual infrastructure, the deployment infrastructure that we got going is pretty simple. It is designated in certain logical zones from public facing down to internal. Simply because we wanted to make sure that we had flexibility in how we did IT. The system itself is on to two virtual machines, they're both 4-core 16 gigabyte servers. They're Windows Server. So we are Windows house and Microsoft Stackhouse. And we use Azure to host all of our virtual machines.

So those two machines, they're both the same. They're 4-core, 16 gigabytes machines, with SSDs, and attached storage for backups of the database and for elasticsearch. So the front end is only accessible internally, but the data we can get through the Heritage Gateway, which will be querying elasticsearch.

OK. So these are the six areas that we tie-in. So it's what I just showed you. And I'll go through those one by one. I will try and cut out the presentation, show you a little demo. But there's quite a lot to cover, so it may be a bit brief.

OK. So one of the big bits of work we've be doing over the last couple years between the two versions that we've released is all around accessibility. And as a public body in England, we have to adhere to a law that came in 2018 around public sector bodies, Accessibility Regulations. We need to ensure now that all of our internal and external applications are accessible to people with disabilities. And where we do that, really, we need to do reasonable adjustments to any applications that we do adopt. And those requirements are really centered around WCAG guidelines.

So we have to adhere to the 2.1 WCAG guidelines, AA, or greater level of accessibility. Where we don't, we need to provide a statement and a road with a roadmap of when we will become compliant. So that's a big bit of work that we've been doing. So we need to make sure we meet

those requirements. So I will attempt to pop out of the presentation and actually show them. So see here we go.

OK. So this is the landing page. What I was going to just briefly talk about before diving into it was, one of the big bits of work that we did up front was going through the whole site and just ensuring that the HTML5 limitation was semantically correct. So just ensuring all the HTML5 elements were correctly used and correctly structured in the right order.

And we also had to make sure that the whole site had been through W3C validation because there were a few elements that weren't. The main reason for this was to enable things like screen readers and the tools that are used in the browser to be able to correctly navigate through the document and parse the HTML. Where it didn't conform, sometimes it would cause issues with those screen readers and tools. So things like labeling, area labeling, just going through and making sure, double checking that they all had them. Making sure the forms, and labels-- so the input things are labeled correctly. So that we can navigate through, and the screen readers could see what was happening.

Some of the tools that we use, things like the Wave plug-in that's a really good tool to just identify areas where there may be things to have a look at, and at least to correct. I mean, I will say when we go in that we've not finished this by any means. We've definitely made a massive inroads into it, but there's still some way to go. OK.

So welcome to Warden. So one of the things that we are quite keen on being able to do, and certainly we've had some internal users that have questioned this, is the font size to be able to zoom and making sure that it's a responsive site. So I'll just give a demonstration, that we can now-- as things increase in size it tends to resize a little bit more as expected. So it's easier to follow, and also it means that the screen readers are still able to go through in the correct order. Like I said, there's still some work to do. So certainly here it's quite a complex page, which makes it a little bit difficult to show some of the results.

We have got a little module that we've used for previous exports before. We had the exports in 5.1, we had a little tool here that we added, which allows us to get a slightly better view of it when we are zoomed in. One of the parts of accessibility is that if what you're using is inaccessible then you need to be able to provide another means for doing that. So this fills that gap while we're still looking at how we can work with that search interface.

So the other things that we've looked at is to do with-- also with vision, you will see actually it looks quite different. There's quite a few differences in the coloring. So we've had to work quite a lot on contrast ratios. Contrast ratios is quite an important thing as well for partially sighted people. Being able to differentiate between the different colors has been quite important. And also things like just little indicators to show a little bit better what the state is, sometimes some of the color contrasts make it quite difficult. So just having a little indicators, things like that, is important.

Another thing is also navigation. So not being able to use the mouse is quite a big thing. So we need to just ensure that that keyboard support is in place. So being able to navigate around, making sure you can get to all the various parts of the site, and open them up. So yes, just be able to navigate into trees, up and down trees, was a good bit of work. I pulled it-- did a lot of work on this, going through and making sure that worked correctly. So I'm just looking at [AUDIO OUT] I'm not going to go too much into it, but you will get a flavor. It's quite a bit different, but at the same time still provides the same functionality that we're used to.

So the next thing that we spent a lot of time on is just looking at some of the extensions. So data types, and functions, and widgets for workflows of the users that are going to be using the application itself. We didn't look at report templates, cards, or plug-ins at this stage. That was not necessary for what we were doing for now. So we're going to try and jump out, again to just give a demo of what we were looking at there. Go away Mr Popdown there. OK.

When people go and develop-- sorry, create new records, the first thing they tend to do is they tend to use a British National Grid coordinate, which we have here in England. So what they wanted to be able to do is put that in, and do a number of things, and prepare a number of the records without-- [AUDIO OUT] the cost without having to go and do that separately. So for that what we've done-- it is-- if I just go coordinate. We can go in here, set up a reference. So you can see here is the widget for our British National grid point. We can answer that in. It does a bit of checking. There is a function there to convert that. Which is done in the view model, and then allows us to create the record.

So what it's going to be doing in the background, it's actually going to be running a number of functions that will generate geometries, and also do a lookup against one of ArcGIS online services to populate some of the data. What I do have to do whenever we run this is, I do actually have to refresh the page because it doesn't update the tree here. So you can see now we've got a point, it's gone away and it's drawing the point, which is relatively straightforward. We've got the grid coordinate there, and also we've gone away and picked up some of the administrative boundaries that intersect with that site.

So what often would normally happen now is the user would come in and they would look to modify that feature, put in another polygon, or something a little bit more detailed around that. OK. That's helpful. Seems right now. Ah, around there. So save that. Update that. What it would do-- if there was any change to the grid reference, it would update that. And similarly, if there was any other intersection with related areas, if it was a larger digitalization, then that would update.

The final thing that we looked to as a function for now is we have an internal indexing system for our records that we're maintaining from AMIE. So whenever a new record is created, it will go and generate this card for us and look up the next index reference in the line. So that's a really brief sprint through that. So I'm going to dash back to my presentation.

So I'm really going to glance over this relatively quickly. One of the things that we have to do is obviously for the Heritage Gateway, provide a public interface into our system to do that. And in

order to [INAUDIBLE], we developed a .NET core service that allowed us to do that. So I should really describe it, the gateway is a way of searching numerous discrete data sets around all of the local Historic England-- sorry, heritage environment records in the UK. So you can see here, there are a number of them. A lot of them are national, but a great many of them are local record. So you can go and type one search value, and you'll go off and search them all and collate those results.

So we've created a new one here for Warden which is this one here. So we've implemented that service. It's a SOAP service. It's a standard interface through SOAP. But what we have done is we've built onto that an additional interface which I was going to show, but I'm looking at the time. So I'm not going to go into it too much, but it allows us to search that, integrate it, and use elasticsearch to provide our data.

So again, in order for us to maintain and manage the system and work with the operations team to make sure that it's up and running OK, what we needed to do was ensure that we had good monitoring on it. And for that we turned to the implementation that Microsoft are using with Opencensus, and more recently Opentelemetry. And this will allow us to be able to look at the system as it stands, how it's running. We can look at logs while it's running and collect some really useful information to help inform us as to what's going on.

So for that I was going to very briefly show how we implemented that. It's a really straightforward process actually. It's also instrumentation. So it really is a case of just effectively applying the packages to your application, and then within the settings.py File, you simply configure the middleware which is here, which looks at the requests. You bring in some integration, which looks at the logging and make sure it transmits the logs, transmits the request, and the traffic, and also transmits information about the PostgreSQL requests. And then simply configuring your logging formatter there.

That is all we needed to do. And just from that, we were able to get a huge amount of information. I must say at this point it's really important that if-- for this, because it uses the logging, it's important not to use print statements in it. And I know, we've gone through, and had to change a lot of our stuff to use logging. And I've seen various moves to making sure we use that a bit more.

But what that gives us is just from simply that bit of code, we're able to see what's going on across the whole application. What traffic go between what dependencies. Where there's issues, you can see here there's some few issues happening between certain elements of the application that allow us to tie-in any information that's been raised by anyone on the business side. And that again, we can go into that a little bit more, we can look at some of the server response times. How the operations are working, the server requests timings happening, you can see in the graph on the left that we've got a few spikes.

So something is taking a little bit of time which might lead you to think that there's something going on. And if we look at any of the logs on the right, you can see that we've got a slight issue with one of those functions. Something's obviously happening there that is causing this. So we

can go and look at that with a bit more information. I've come away from the presentation that's not what I want to do.

So for the last two things I was going to talk about, there was a great deal of-- there's a need for us to develop resource views. And for those of you who aren't familiar with resource views, it's a way of us to write into the database. So views flatten that graph structure that the Arches uses underneath in order to generate some database objects which we can plug into for integrations.

We actually spent quite a lot of time doing this because we could see the benefit in some of the things we were looking at. Some of the other systems we have that could allow them to connect in fairly easily to the standard database objects, and pull out the data. We did have-- I'm not going to show you what it looks like just for now, because I'm conscious of the time. But we did generate quite a few, it was about 90 in total, which is a lot. Some of them are quite simple, some of them are more complex. And they provide information for the things I'm about to show now.

So I mentioned earlier about bulk reporting. As I said, we get contacted to get information about what's in the National data, and sometimes if it's very big, but it could be of any size. But if it's a particularly large request, then we need to be able to provide huge numbers of reports and data to those third parties.

We looked around, we didn't have much money for actual reporting solutions. So what we decided to do actually was to develop [INAUDIBLE] workflow in Talon that allowed us to very quickly tie into all those various elements of the models, generate an HTML report. And the way it was written allowed us to customize that very quickly. So we can remove and add in that information very simply. And also we can have multiple types of report for the different requirements that we might have with reporting.

And so really onto the last bit that I was going to talk about, and that was the ArcGIS tool integration. So at the moment, I know it's a work in progress. But at the moment there's no standard connection between the two systems. We predominately at the moment use ArcGIS template 3 and 4. We have got an upgrade grade path, but at that point there's not there wasn't a whole lot that we could do. So what we have done is we've used those materialized views to be able to generate views or fairly simple views of all of the spatial data in the database. That allows us to, for the desktop solution, stream out with a direct database connection into the desktop environments. So the users can access that data fairly rapidly.

It's quite a lot of data. So feature services with ArcGIS don't really-- sometimes they don't perform that well using this approach. So what we have is an additional process for exporting some of the spatial data from the views and distribute it around to our various systems.

And then just over on the left here, there are a number of workflows that we have within some of our teams that require us to be able to join some of the searches that we do within Warden, and have that influence some of the spatial data selections that we're doing.



And really, the way we were able to do it relatively quickly is to get on the back of the export, the CSV export. So in the early version we wrote one that did it for us based on the existing search, but later on with version 5 it comes with the export functionality. So we're able to do a search, that rich search within the ArcGIS user interface, export out a CSV of that search that represents it, and then use that CSV to influence what mapping we're able to then select and potentially enhance with some intersections, or if someone has a polygon that-- it gives a specific search area. We can then integrate that and create a spatial data set based on that.

And some of the other tools are a little help [AUDIO OUT] such as if we need to do some fairly complex digitizing of polygons, we can do that here and export ArcGIS, some representation of that for importing into the record. So if we can't do it within the user interface of ArcGIS, that allows us to do it.

So that was fairly rapid. I do apologize I did talk quite quickly, but there's a lot to cover. And it's just glancing over the surface of it. But yeah, that's pretty much everything. So I've just stuck within. I've got 28 minutes here on my time, I'm quite pleased with that. I guess that will be the questions now.

Andy that was really great. And you covered a lot of ground, some really interesting material. And thanks for the presentation. There were a couple questions in the chat section, and one about accessibility that was answered by one of your colleagues. There is a question outstanding though, and that is about the data wrangling bits and the tools that you used, you and Shantelle used to kind of wrestle the data to the ground, and that transition, I guess the extraction, and the processing from AMIE into Warden, could you just spend a moment talking about that?

So yes. FME was the main tool that we used. I know that the heritage data team who did some of the cleansing work used OpenRefine, I think for a number of their tasks, but we really relied on FME to do that. We found it was far more flexible, we could very rapidly change the workflows to allow us to extract from the tables, reorder it, go in and manipulate it, use it to build JSON structures and CSV files. And it was also really good because it had full spatial data support. I think its origins are in GIS world. So it covered both those areas in terms of being able to pull out all that data and also do the spatial data transformation.

And it was pretty fast as well. We were dealing with a lot of data, and it had to go through a lot of processes to do it. So being able to parallelize it and use fairly beefy machines to do it was really good. So it was a good experience, if not long and quite laborious on occasions.

Yeah. Well, underestimate data manipulation and data transformation at your peril.

Absolutely.

If you are going to remove something from a legacy system into any new target data management system Arches or whatever, it's a lot of work. And 20 years of data management on AMIE makes it that much astounding.

Absolutely, having what is effectively in there about six different databases shoehorn together, trying to work out how they all join up. And make sure that we're capturing the data that we actually do need, leaving behind the data that we don't need, because it's lost its meaning in the histories of the mists of time. So that was quite difficult. We had to do that process quite a bit. So we would pull the data out, check it, find it was wrong, have to do it again. And having that all, the flexibility of very quickly spinning up different sections, different checks, it was really useful.

That's cool. Just in the interest of time, I'm going to suggest that we move on to our next presentation, which will be something I've been looking forward to, the conversation about Arches, and Docker, and Kubernetes, and really, I think, a nod towards thinking about Arches in production. This is a presentation from Sarah Bryne and Phil Weir of Flax and Teal. And I think Sarah's going to start. So Sarah I'm going to make you the host.

If you could make Phil the host, because he has all of the presentations. I'm just going to do a bit of an intro to some of the instances.

OK I will do that. Let's see if I can figure out how to do it. OK. Let's see here. Phil are you able to share your screen?

I could go here, I'm stuck. Find my screen first. There we go. OK. Is that visible there?

Yeah that's perfect.

OK. Great. So yes, as you're saying. thank you very much. Sarah's going to start with a little bit of an intro. So I guess, Sarah did you want to start with the background and then I'll get the sites up as you are talking?

Yeah, that sounds good, thanks very much Phil. So just to give you a brief overview of who Flax and Teal are. I'm sure many of you haven't really come across this yet. We're an open source focused software company based in Belfast, in Northern Ireland. And we primarily work within the data management and engineering fields internationally, but we thought that the way that [INAUDIBLE] platform uses data really suited our skill set. And we're really excited to be part of the community.

So I'm Sarah. I'm the product manager for Arches. I actually have a background in medieval academic history, primarily in the history of medicine. And I have been a volunteer for a number of the cultural heritage organizations here in Northern Ireland as well. So I feel like I understand at least some of the patterns in some of the organizations. And so I was very keen to just thrust this forward.

So we have on the call as well Phil, who's the director and founder of Flax and Teal. He has background's in mathematical modeling of sea ice in the Antarctic. And he's also a very early adopter of Kubernetes, which I'll hand over to him to speak about in more depth after a quick look at our current projects.

So Phil if you could just go to the islands and spaces.

Yeah, sure.

OK. So I'm not going to go into a lot of depth. You'll know what Arches does, this particular instance which is on Kubernetes infrastructure was an instance that was set up as a demo for our customers in the Republic of Ireland. So all of the data on this is open data at the moment.

So it was a testing instance really for me to alter the metadata and to work with customers to show how some of their assets could fit into the Arches framework. It's currently using the CIDOC CRM at present, but we have been working a lot with assets that have been relocated from one part of Northern Ireland to another. So we're working that into the metadata as well. As well as shipwrecks, which is going to be a really big part of one of our customers' data sets.

So another one of our instances that is particularly interesting is Anthropos. So this is very new. We've only really just got started with this one. And so again on Kubernetes infrastructure. It's an instance set up for the New Zealand Antarctic Heritage Trust. And I'm currently working with them on their database to advise on the CRM going forward. So I'll just hand over to Phil for the main event. I know that you'll want to know about Kubernetes.

Good. Thank you very much, Sarah. So I'm going to apologize, it's going to inevitably be a little bit cloudy. And I know that's not necessary, pollutes everyone's boat. But it's a wonderful pun related segue. Oop, sorry. [INAUDIBLE] One second here. [INAUDIBLE]

Oh, that's right. I moved of the presentation to show that. So if I go back to where I was here. So probably a more useful title in the [INAUDIBLE], but I thought maybe to try and capture the flavor of what we're doing. We're trying to use some of these tools that we've got to experience using in other settings and seeing if we can get some benefit with pairing them with Arches, and if that's something that's useful to other people too.

So first question, and that some of you who've been watching over the last probably half is what is Kubernetes? And I'm aware that some of you would have come across this, and actually used it, some of you have probably heard of it. It is obvious [INAUDIBLE], which we thought. So it's one of those kind of buzzwords that tends to come up a lot in the industry generally. So the other question I'm going to try and answer is why that's actually relevant to Arches? And I'm going to start with the first. And try and do a bit of a whistle stop tour of what Kubernetes is and to get people at least able to see why we're having certain conversations, or what the chat's about.

So key practical points with Arches. First of all, it's well standardized and open. And that's one of the things that we really quite like about it. There's a number of ways that we can use this type of infrastructure. And I guess I should really clarify that the main purpose that we are looking for in Kubernetes is trying to run processes in the Cloud. I think that's really the key concept. And there's lots of ways of doing that. And that could be setting up VMs manually, it could be setting up the VMs automatically, could be running Docker Compose and Cloud on AWS or Azure, or something like Kubernetes.

So what is it that attracted us to that? It's well standardized and open. This is something that from the outset was trying to solve some of the challenges of having very different environments people are having to deploy tools like Arches onto. It's something that lets you get to that next point relatively standard way. So to say things like, I suddenly want to think about high scale outs across multiple machines. How do I do that?

And part of the idea of Kubernetes is that there's been a brick wall in some ways. And traditionally you start having to think about operations and Cloud ops and things like that. Where you really want to just think about how you put together a bit of software, how it works? How it interacts? And not be having to deal with Azure management on a case by case basis. So it makes that flow a bit more straightforward.

It's also something that's quite well structured. It did come out of Google originally. But it's been handed off. And it's got quite big buying in from most of the major IT players, particularly Microsoft. And it's something that has been relatively widely adopted by organizations that don't tend to work together. So for example, Amazon, Microsoft, and Google.

And it allows us, and this is something particularly-- and we're an SME. So for our benefit, having our operations processes, things like how do we set up a new stack? How do we get a series of VMs up? How do we get load balancing work? That sort of thing, it keeps it quite standard.

And there is a perception that Kubernetes can be quite complicated coming to it for the first time. We've actually found this standardizing benefit means that we've been able to use that with our junior developers as well, and get them up to speed, and get them being able to do basic functionality too. And one thing that's from a very pragmatic point is it's increasingly standard for a number of government departments. And it's something that we see come up quite regularly nowadays in terms of opportunities and so forth. So it's a basic expectation. And GDS are quite big enough. And I'm aware that that's not restricted to the UK either, lots of places around the world that's the case for them.

And on a technical level. So YAML, you'll see a few examples of this going forward. I'm sure some of you are very familiar with it. Some of you may not come across this so often. But it's a consistent way of us having documentation. And it is a superset of JSON. So effectively, things that you can represent in JSON, you can represent in YAML. And it's a little bit easier to read. But that tends to be the definitions that will turn into an actual running piece of infrastructure testing YAML.

This is something that works with all the major Cloud vendors as I mentioned. But it also works on VMs, and it's something that you can in fact, if you've got a bunch of machines, you can even set on that. And there's a lot of work at the moment going on just to have even Raspberry Pi use these deployments, where you can fill your house with Raspberry Pis and have them all operating together in this way.

It works quite nicely with standard flows. So we tend to use it a lot with GitHub, but also works well with GitHub, and other FCI solutions. And that means that you change a piece of code, or developer changes piece of code, and the next thing they know it's running live and there's a nice audit trail and it's followed through and gets and knows how to do any manual steps and which makes it obviously much harder to track and audit.

And in that case, two key points of this, one is having everything recorded so that there's no ambiguity as to who is doing what and what infrastructure there is up there. The other is you don't want to have to think about servers if you're a developer. And if you're hitting limits of servers, we need to scale past them. Kubernetes abstracts that away. And I can talk more about that in questions, if that's helpful. But we'll come through that as we're talking.

And it's also great for complex scripted tools. So we often tend to-- Python-based processing of let's say maybe a machine learning algorithm that we want to deploy alongside a web framework, or potentially a simulation or engineering process. And Kubernetes gives us a nice straightforward clean ways of saying, here's five things that need to put together. And here's the number of each of them that we want, here's how much priority to give to them and so forth. And that's all encoded and not things that we're having to manually do.

I highly recommend looking at this. It's a little bit old now, but it's going to be much more visually engaging than I suspect I will. And it gives a walk through-- it is an actual series guide to Kubernetes. I love it. It's fantastic. But someone sat down and thought, well, actually, how can we take some of these ideas abstract them and put them into a nice walk through discussion. And it's interspersed with technical drawings. So if you're thinking of Christmas presents for your kids, then you probably want to take those pictures off.

I'll only touch on this one relatively briefly, mainly so that if there's any questions we can come back to this, you know it's here. But along the bottom, we have these nodes. These nodes are VMs. They might be VMs in Azure. They might be VMs in AWS. They might be on your computer. And Kubernetes looks after everything on the bottom left. So everything on that one side of that dotted line, it looks at how to put it all together, and tries to obstruct those problems away from you as much as possible.

On the top right, you have the processes that you want to run, you tell Kubernetes and it just works out how to run them, where to run them, and how to keep them up. And if they start looking to start looking sick it'll try and heal them. And if they fall over, it'll bring them back up.

If you're wanting to play around with Kubernetes, [INAUDIBLE] as minikube, you can run that on your local machine. And it gives you a full featured Cloud-like deployment, but you can run it in one virtual machine on your desktop. Experiment and see how these things go. We haven't tried this with our Arches deployment yet, but interested in feedback if you do.

And the final bit of what I want to mention is called Helm. And this is the magic bit where Helm is a way of sharing templated applications. And that could be PostgreSQL. Or it could be ckan, for those of you who know ckan, could be a particular application, but a full SaaS application.

And we have pretty much all of our projects, we will have our infrastructure definitions in this format.

And that means that we can give it specific contextual values, and all of the things it needs to know, what can talk to what? What security things like between these two processes? What's a number of processes we want? And of PostgreSQL, or of Redis? How many different storage containers we need? That sort of thing is all stored in that template in a way. And we can fire up three development environments very quickly for example.

So bringing it about to Arches. And we've got some first steps here. And you'll see that we've got a series of hopefully familiar looking names. This is as I said first steps. We're just running most cases one instance of each thing. And key notes to point out, after the elasticSEARCH couchdb, we've got PostgreSQL server running, that would be running. PostgreSQL and we've got RabbitMQ alongside NFS server for handling storage so we can upload things to this.

And Arches, this is really an important point, and Kubernetes pushes us to try and make as much of what we're doing stateless as possible. That means we can effectively scale it easily. So here we have Arches itself, and we've got our static assets in a container, that's something we want to change. But Arches itself and Django is stateless. So we can scale up, and say to Kubernetes actually we want five of these, go off and make that happen.

Just to demystify a bit of what's going on in the background there, at Nginx as a simple example. We have, I mentioned YAML. And this is a boiled down version where we stripped out some of the boilerplate some of the standard stuff. And I think this is almost enough to get a web server running in Kubernetes. So we have a pod which Kubernetes takes quite a strong wheel based analogy for various reasons. So a pod-- what we have are in this case one container, it will run. And we have a service which encapsulates the idea of, OK, maybe I've got 5 Nginx instances running, and I want to load balance between them. So services called Nginx and Kubernetes are magically redirect to each of these individual containers.

I also mentioned that Kubernetes can self-heal. So how does it do that? Well actually, what we have is this idea of a deployment. When we create a deployment, we maybe have one, two, three, four, or five replicates, or maybe we have auto scaling, even better. And we say each of those will be able to stand up containers that look like this for those of you who are familiar with Cloud Infrastructure. And that might be like an auto-scaling group or something like that.

And that's enough for us to say, OK, we want five Nginx instances that load balance between them. One, three, seven and effectively these two bits of definition are enough. Of course, we can store that in Git, which means it's versioned. And we can even if we really want to put it into disintegration, so that every time we change it our Kubernetes app cluster as we call it, will automatically update. So if we wanted to change something about our definitions we push it to Git, and the next thing we know there it is on our development site, or live site if we were to dream.

Just because I'm aware we need some pictures in this. I've got a picture here of some of what I mentioned. Helm is a bit like a package manager really for Kubernetes. So there's a view of the Helm charts as they call them, or packages, that you can fire on to Kubernetes cluster. And effectively, you give it a set of values and you say, OK, I want to know-- what we got there, artifactory instance, or airflow for those of you who do work flows. And you'll give it specific values that you want to customize, and all of the actual structure of that application that's contained in that package. So you can launch them pretty quickly. ,

And that works by pairing your configuration, which might be quite short. So effectively we could say, OK, let's bring up a series of virtual instances, each of them will have a short bit of config, and then all of these standard bits of YAML will say, OK, we want Django running here with these variables. We want Redis running here with these variables. It'll put those together and turn those into an actual running thing on the Cloud. And say for example the Azure AKS.

I've mentioned continuous integration. I just wanted to highlight that. That's something where we can push it, test it, and see it come alive. But in particular with Kubernetes, it gives us a way of when we push it by default, or change the container that it's going to put in. So this is all Docker containers and [INAUDIBLE] here.

So once the Docker container is built, GitHub, for example, will do that without really any significant configuration. And it will push it to Kubernetes, and that will do a progressive rollout. So we're not even thinking about downtime, and wait until each of our instances is stable. Each of our processes, or replicates, or whatever we want to call them are stable before closing down the old version. So it's sensible enough not to switch everything off when we upgrade, but to let it happen progressively.

So a few considerations. Again, this is something that probably is going to be picked up in discussion. If not this evening again. But creating new Arches projects, we want to have a bit of a think about that. We got a flow, as you've seen we've tried this with two, and we were even able to get a 5.1 up within 24 hours I think it was. We were running our 5.1 upgrade. And that went largely fine, a few issues but generally fine. So we're getting enough flow going. Looking at how we handle static assets in a way that fits with Kubernetes without making things complicated on the back. end.

Storage, I should very briefly-- NFS there. There's lots of different ways that we can deal with storage. We also use [INAUDIBLE] in other circumstances, which allows-- it's a bit like Azure. Storage containers right up to its buckets, but for the moment we're using network file server. And that does the job nicely.

We have to think a little bit about DNS, because obviously something's got to point into this. Kubernetes does a great job of handling that in Wordbit but we do have to think how do we point our site to the right bit of Kubernetes, or at least at the front door of Kubernetes. And then we have to instruct it where to go from there,

Autoscaling, something that's-- if we've got different types of metrics coming out of an application, we can say, OK, now we need to have 10 processes instead of five. And that's not something that we have to do in a very simplistic way either. So that's something that's worth discussion.

Maintenance tasks, this is one that we've had a few times where we need to run a command. Kubernetes gives us ways of basically running a command on an individual process and jumping in and saying, OK, I need to run this Python task here, but that's maybe something that needs a deep dive at some point.

Backups. So where do we put our database, for example. How does it get synced? Why is that getting backed up? Where's that getting backed up? How do we give a sensible configuration that if someone wants to use it that they aren't having to think about that as an afterthought? And upgrading, so I mentioned we did the 5.1 upgrade very quickly, which is a positive sign, but we want to think about it a little bit about that. That's it. And I'll see where we can go from there. So I'm going to do a quick summary. Sorry, I missed the start there. Dennis, how am I for time here? I guess that must be near half an hour.

Phil, you've got a few more minutes though

OK. Great. Thank you. So I'll do a quick survey of some of the code. This is still quite rough and ready. And so we are iterating. But the key points for us with Kubernetes is saying why do we want to insert this layer between, for us at least between, Arches and say Azure or AWS or--

One is the openness. We know that anything in Kubernetes, the whole stack is open source. So it means that if we need to jump into the code to try and understand something, we can, which we have to do and asked. Something odd happens or a particular process. I've been recently having fun trying to resolve some issues with rollouts from major cloud vendors, where we don't have insight into that, and so that has actually pushed me back on at least one project to choose in Kubernetes

Scalability. As I mentioned, it forces us into that thinking of having stateless processes where possible, which were quite easily scaled up as users increases. Makes popping up lots of instances of say Arches much easier, I think. In that we can say here's our template bam, bam, bam, bam, bam. Here we've got a whole load of them. Now, there is a question about how we deal with creating new projects, and that's something that I think will be an ongoing discussion, hopefully on this. But in terms of we want to bring up a new version of this with maybe certain environments variables changed or certain conflict changed becomes quite quick.

Resilience and consistency are quite closely linked. Having all of the infrastructure definition as code means that I can set up let's say go into Azure portal, there are [INAUDIBLE] too, but go into Azure portal click AKS, new cluster, and then fire this onto it. And everything else is in code. And everything else is in Gits, everything else is versioned, it's recorded, it's auditable, it's traceable. And if it goes down, Kubernetes has all of that code to tell it how to bring things back up. So it's been fantastic for those.



OK. So enough science, I'm going to attempt to find my screen, here it is. And let's see, little bit here. So this is on and GitHub. As I said it's a little bit rough around the edges, so please do bear that in mind. But under our ...Flax & Teal report.

What we've got here is an Helm chart. An Helm chart as I mentioned, is a template. As a template it's got a bit of definition information, so we say, OK, this is an Arches project, and an Helm Chart, what's it for? A bit about version, that kind of metadata. That's all recorded somewhere. And then within that we've also got dependencies. So how do I get RabbitMQ running? Well, I put it down as a dependency. And just like we were in Python, or a little bit more variable setting, but as if we were looking at a Python dependency or JS dependency, we can pull those in we're no longer having to [INAUDIBLE] separate, how do we set up PostgreSQL kind of plans.

OK. And within the tablets we've got some of those things that I showed earlier. For example we have config that will pull in into environment variables, so our config settings. We can have secrets, obviously some things we don't want to be visible in the normal config, such as db passwords and so forth. Those can be injected as files. So as far as the running process is concerned, it just sees another file sitting around the place, and its content that it thinks it's properly running on your local Docker, or a VM docker, but in reality, it's much [INAUDIBLE] by Kubernetes.

And we can also have our workers running. So that's something where we can do, say well, workers have grown jobs. And these aren't all web serving processes, we can have this in, and like I said templating config. I don't think I've really got time to go into syntax too much, but at least so that you can see how some of those things are linked together there. For example, we're pulling in couchdb password. And we're pulling it in from our standards, from a secret. And Kubernetes will keep track. When that changes, and the couchdb password will change as seen by Arches.

So we can even do things like look at how much of what the resources being set, given to each are. And that will help Kubernetes decide if it's got six VMs, for example, available to it. How it distributes all of those processes. We don't need to worry about that. We just need to say this needs half a gig, or this needs 100 Meg, Kubernetes will magically put it in a sensible place.

So we've had to do a little bit of playing around with the Dockerfiles and project creation. Oh, sorry. [INAUDIBLE] go back to questions there. So some of that, we've got a couple of extra bits for pulling in entry points and making sure that we've got everything that's necessary within that container. One of the things that we do like to do with Kubernetes is make sure that we don't have time for [INAUDIBLE] and just as an safety layer, so getting any of that conflict set up, and making sure that it's fully able to get all of its [INAUDIBLE], either from files or from environmental variable.

And the rest is fairly standard, nothing too dramatic. And the next thing we know we've got something that we can build fresh to Docker, Docker hub, or say for example I should get a registry, and Kubernetes will happily put it in, and run for us. So I'm obviously happy to go

spend more time on the code, more time on the questions, more time on the future bits. There's a couple of questions in the chat as well, but I'm also aware that I'm probably pretty tight for time. So I thank you very much. Dennis, I'm happy to take any questions if we have time at the end.

Great. Thanks to all that. That's super cool. And I appreciate you going through in such a nice organized fashion. It's really super interesting. There's I think and also an interesting question from Adam Cox about Ansible. And Adam, I guess I'll leave it to you, do you want to-- since you're the next presentation, would you like to do your presentation first and come back to this question, or would you prefer to get the question out of the way and then just move it to your presentation?

I would love to hear the answer to the question while this presentation is fresh on everyone's mind.

Great. Maybe, Phil you could do just a quick-- give a quick little summary of whether you looked at Ansible or not, and if you didn't do so, what your thoughts were?

Yeah, sure. So I guess Kubernetes fits somewhere a little bit in between. There's a bit I suppose it combines some of what Ansible does, but not everything with some of what Docker Compose does, but not everything. And in some ways I feel that's underselling it, but I think it's the way I'm thinking about it. What Kubernetes is trying to do is say, you don't want to be thinking about individual and operations assets. So when I'm dealing with Kubernetes and everything that I've got to find here, there's no reference to what type of VMs, or what scale of infrastructure or anything like that.

Kubernetes is effectively saying, how do I get this series of processes to run. And everything else is either someone else's problem, or something it can do under the hood. So I've used Ansible in the past, I do actually like Ansible, but one of the combinations that I've used is Terraform and Helm. And I think that give us an idea that Helm, what we're looking at is the app structure, it makes sure that all of the app components are running, it doesn't talk about what components in Azure, say, need to be running. And for example at the Terraform end or the Ansible end, you can make that explicit or you can leave it to do things, but Kubernetes works that itself.

So for example, if I say I want to have an ingress, so it's I want this particular service to get an IP address. That's all I really need to tell Kubernetes, and all of the actual fine detail of how that happens or who has to set up load balances, or whether that load balance needs to be connected to a certain VPC, or according to your Cloud provider. That's all handled by Kubernetes under the hood. So it's really about trying to-- I think of it as doing all of the building underneath, that you can have a nice flat level surface. And that level surface is the same whatever platform you're using. There's no thinking about operations, it's declare what processes, what processes are running on, what infrastructure are running, does that help a bit?

Yeah, it absolutely does. I think I've explored using Ansible a little bit with Arches, but I'm seeing it fitting in a piece of what you've shown. But this is a more complete ecosystem I suppose, would be the way to put it.

Yeah. I certainly know of people who use Kubernetes and Ansible together. And so there are things like, say for example, we're working on IoT solutions as well at times. And you know there's a lot of stuff there that maybe we want to bring up IoT hub in Azure or something like that. That isn't really something that Kubernetes-- again part of the point of Kubernetes is that it's focusing on a cloud application. And so something else is bringing that up. So it can sometimes be in partnership.

I think something like Arches we haven't had any issues using Kubernetes on its own, but I suppose something creates a Kubernetes cluster. And that could be clicking what? Three or four buttons in Azure portal, but equally that could be Ansible or Terraform. And obviously if you're wanting to have production infrastructure, then I guess you want to have the Kubernetes set up, also that infrastructure's code. And that might be a small bit of Ansible or small bit of time from doing that.

Great. Thank you.

Thanks Phil. If you could stop sharing your screen. And Adam if you wouldn't mind taking on screen sharing and get yourself prepped for your presentation. Just by way of introduction, Adam is going to be talking about what he's done with a group in Florida. And we're back to a presentation about Arches as a particular solution. And I've been really looking forward to hearing the details of this for quite a while now. So Adam, thanks for offering to share your work. And please, take it away.

OK. Great. Thanks everybody. I've been excited to present this. And it's extremely daunting. There's so much to say about it. So I'm going to try to fit in the amount of time, and also give a little bit of the interesting details about this project. So the program is called Heritage Monitoring Scouts and it is for the Florida Public Archeology Network, FPAN.

A little about myself, I've been working with Arches since summer of 2014. And I've done a number of different installations for different folks. And I'm currently at school, getting a master's degree at LSU. And this and one other project are the only things I continue to work on, substantially with Arches, at the moment.

Here's a quick overview of the presentation, I just want to give a little bit of background about the project, and then some of the characteristics and custom features. And then I'll dive a little bit more into how we handle archeological sites. And then some diagrams for development and production workflows, which are, I think, pretty basic, but hopefully that will be interesting to people. And then just a quick wrap up of what the current upgrades that are planned.

So this Heritage Monitoring Scouts, or HMS program, is like I said part of FPAN. You can go to the web, fpan website, just [fpan.us](http://fpan.us). Florida Public Archeology Network, I can't give a fair background on it because I don't know all the details, but I would say that few states in the US, I think, have such a great non-profit entity like this. And they're a cluster of different offices around the state. And they do a lot of great public archeology work. So I do recommend looking them up and learning more about different other projects.

Here's the URL for our Arches implementation, is [hms.fpan.us](http://hms.fpan.us). And the underlying program here is to use citizen engagement to get condition assessments of historical sites around the state, especially with regard to sea level rise. Obviously Florida, lots of coast. And additionally they were very insistent, as you'll see, on making this something that state employees can use as well. And that's worked out great. A little bit of details about the database, we built the project directly from the existing historical database from the State Historic Preservation Office FMSF Florida Master Site File.

And so there are six different types of resources in that database for the HMS program. We took out three of those categories. And these are the approximate numbers of how many resources there are. And this is-- I also have some complicated filtering to take a very small portion of the historical structures from the state database.

So we basically pulled all of those into Arches. And then we have a fourth category in Arches called Scout reports. And these are condition assessment forms. And as you can see, in Arches lingo this was the creation of four different resource models. And I used, since the beginning, resource instance value or nodes to start reports with-- well not quite since the beginning, but to connect scouts reports with the archeological, or cemetery, or historical structure site, that the report concerns.

So a little background on how the FPAN folks found me was that I was helping in 2017, presented with [INAUDIBLE] at the CAA in Atlanta, Georgia. And a couple folks from FPAN attended the workshop. And this was actually I think pre-Arches for at that moment, or a release on timing, just before the release or the release. So they were really interested in essentially using Arches to upgrade their existing HMS project. They had one that had been working for, I don't actually know, maybe a few years. They used Google Sheets and forms and photo submissions, et cetera. So this was an existing program. And we were going to use Arches as a 2.0 version.

This came with a number of special requirements. These are the ones I'm going to talk about the most. These are the most important ones. So they right off the bat wanted to make sure that there is separate sign on portals for scouts and for state land managers. I would say that getting state manager buy in was really important to them from the very beginning. And so we wanted to make something that accommodated that. In a way it's a very cosmetic, because we just wanted people to think they're signing in through different portals. I mean, they are signing in through different portals, but it gives the site a different feel is what they wanted.

And then we also needed to support public sign up which was something that Arches did not support at the time, but now it does. And then finally, archeological sites are the point, the core of this whole thing. And they required some pretty complex permissions, which Arches now support. Some pieces of this as you'll see, Arches support some pieces of what I'm going to be showing or describing at least. But at the time they didn't at all. So these were new ground I had to figure out. And I'll get into the permissions especially on this drive a little more.

First, in terms of the look and feel we wanted to base this directly on their existing website. So you can see here the FPAN website is very nicely filled out, lots of different regions, and

information about different regions. So build from strength. So we wanted to build from this very existing success and good branding.

So the first thing we did was rebuild the Arches homepage. Significantly, obviously, it is very different. And so we slapped a new set of templates on the front of Arches. So it looks pretty native to their existing program. And you'll notice here there's two different sign in portals. This one on the left, it's a scout portal. On the right-- I really should fullscreen this, but I had some issues, it didn't work. But I hope you can see well enough. But anyway, two different sign in portals, and going to those portals gives a preview of the different CSS styles that we set up to permeate through the app once someone has signed through that type of user account.

So again, it's pretty cosmetic but it was a very important aspect of the whole project to prioritize from the beginning. What happens really on the template side is that a different CSS file is loaded based on who is requesting the template. And I now have thought of many better ways to do that. But this is what we have at the moment, and maybe sometime down the road I'll be able to refactor that a little slicker. So when a scout is signed in, this is what it looks like. And just again, pretty basic CSS changes, but state users signed in would have different color scheme, et cetera. OK.

The thing I want to give a pretty thorough description of, though it's overwhelmingly fragmented at this moment, is how we handle archeological site permissions. There are a couple of requirements for this. And so most importantly, the public can view archeological sites, and scouts can only view sites that they've been assigned to. This is their existing workflow, so we wanted to stick with that workflow.

And then, additionally like I mentioned we needed to accommodate state land managers. So they needed to be able to see various subsets of all the archeological sites. And this was something that we designed very, I would just say, much to specifically around the initial request. And later I've had to patch on change strategy [AUDIO OUT]

Adam? I think we lost your audio there. Is anybody hearing Adam? There we are.

Oh, I see.

There he is.

A connection, oh no, I'm sorry. That was just a-- yeah, is this slide familiar to you?

Yeah, we've seen this slide. But we lost you towards the end of your discussion on this one.

OK. Well, then you didn't miss much. At any rate they're just different state land manager types of access that needed to be handled. So I wanted to figure out a way that could handle both of these requirements, of course, simplify as much as possible. So what I came up with was building around the strategy which applies permissions based on user information compared to node values in the resource itself. And ultimately, at its base this has been a really good strategy,

though there's plenty of extraneous things that still need to be figured out, well not figured out, but fixed up. I would tell you they we're refactored.

But ultimately, this accomplished both of these things because I was able to add a signed to node for resources, or archeological sites, so that admin could change that node to equal the scouts name. And then the filter can just use the user name. And then the secondary version of this was, for the land managers, I would associate location and then management agency information to each resource with a spatial join. And then compare other aspects of the land manager users, so maybe not their name specifically, but the groups that they're a member of, et cetera. And compare those to the values in the location or agency nodes.

Ultimately, that continues to be the strategy. Well, one note here, I found it quite difficult to apply these filters to only one resource model. And that was a lot of elasticsearch wrangling, so to speak. Made it work, but that was a real challenge initially.

So ultimately Arches, as you're all aware, shows search results on the search page, but it also shows tiles or vector tiles of resource locations. And so neither of those can be available to scouts, or the public, et cetera. So two places the search filter had to be used, one is an [INAUDIBLE] search component that affects the actual search results. And then what I've done is pull this api.MVT, I think that's the name of the class, straight into my app. And then really just passed in an extra aware clause to the SQL that runs the MVT query, to pull out a specific subset of geometries based on different criteria.

So this has been working since-- well, for a while I guess. It's changed obviously, the MVT. I had to have a different strategy earlier on, but this has been the strategy for a while now. So I can do it on just a quick show of what this looks like. Oh, that's where it showed up. So I have this running locally. I'm signed in as an admin in this private window, and signed in as a scout C.J. Morton, Chad Morton, example scout.

So one thing to note right away, these are some demo sites basically, but search results are different here even though there's really the same-- I can't move that, the same URLs basically are being used here. One difference is there's a site filter, that's the custom search filter. So the workflow is that an admin here we'll find a site, and then go to edit it. And this is where the admin is able to choose a scout to assign the site. This is a custom widget, not really time to talk about that, but ultimately that pulls from the list of scouts and in the Django data ORM. So now that we've assigned that scout, so if we refresh the page we'll now see-- this guy can now see the site he has been assigned.

There's a lot of other extra things that could be added to this that we just haven't been able to get to. So again, obviously, when the detail-- this is 1408 now, not 1407. If we filter for archeological sites we would see this one. And the scout can now view the report for this site. And then we added a link in the report for the site which says, something like Create Report. Which is really just a-- so Create Report. So this is just a shortcut, really, back to the normal creation method for a new scout report resource.

That's the workflow. People come through. Like I mentioned, there's a lot of extra stuff that could be done. We could trigger-- we should trigger emails so that scouts will know when a new site is assigned to them, et cetera. But there's a lot to figure out just to get this far. And I won't demo the state land manager permissions, but it's a little easier for them anyway, because they are not waiting to be assigned a site, they just can sort by archeological site and then they'll be seeing all of the sites within their park.

OK. So I'll jump back to the larger presentation. So I guess I'm doing OK on time. I wanted to just talk a little bit about some of my standards of practice, I guess at this point for development and deployment. This is probably very, very familiar to most of you. But I just wanted to put it up there to show the way that I handle this type of development. And especially for this project, but this is I guess what I use in general too.

So as you're all familiar with. We have the core Arches code, the project code, and then in this case, the package which I have fpan-data. So these are all different repos. And I guess, I just wanted to show that this allows the exact same code to be deployed to the staging server as well as to the production server. So it's very easy to make the changes locally, test all I want, and then also push them up and deploy it just to the staging server first. This is really important, not only to see what happens once it's up on a server that's not my laptop. But also that I can show it to the clients, and they can have a look at it, and test new functionality test new features.

It's also especially important because we're dealing with archeological sites, and so this is sensitive data, and we have not had any problems with it. Of course, knock on wood, which I'm doing right now. But we've kept it all very safe. And part of that is having a good deployment strategy. So once it's been deployed and tested on the staging server, then follow the exact same pulls in deployments to push it into the production server. It's relatively low tech set up I guess, but it's certainly sufficient for this project.

I wanted to mention just a little bit about the difference between the project in the package. I won't go through all these points, they're basic or known to you. But for a very long time, I really managed the package closely and made sure that it was a very close-- could reproduce all of the content of the database. And for the most part that's still true. We still push changes from resource models back into the package, so that they stay in one place and can be redeployed at any time.

Probably the biggest difference is that we initially had a whole, and still do have a whole set of state user accounts that we needed to make. And I've moved away from trying to make sure that those are always mirrored in the package. I had some complex scripts that would run through your lists and create users, and assign groups, et cetera. But as we're moving forward with this we just need a lot more flexibility with state manager accounts. So I'll talk about that in a little bit here.

But that dropped off trying to manage, especially keeping all the user accounts faithful in the package, but otherwise they still keep everything in there for us to-- at least the base information, and some sample but of course no real archeological data.

This is the current production set up that we're using. It's evolved a little bit over the last few years. I guess I did mention this earlier, but it was in 2017, we started this in the summer. And I think our software release was February 2018. So it's been in production for a couple of years. And there's a couple of things to say about this, reusing the PostgreSQL and elasticsearch servers, really useful, really helpful, additionally because this elasticsearch server is used by a ton of other projects that I have. So it's been nice to be able to reuse those.

I would also mention that we-- a long time ago, again, we were setting up email from this app to accommodate the sign up before that was really baked in the Arches at all. We set up a local server on the production server using PostgreSQL. And I would not do that again. Only because Syndicate is a much easier client to use for managing emails. Other people probably have other ideas to other solutions. So managing an email server on top of your production server, it's not like it requires a lot of management but it's probably best not to add that to the stack.

And then we use S3 Buckets for uploading media like photographs, and then also to store backup files from the database, which run daily. I thought I just mentioned the cronjobs that we use on the production server, frankly we could add a number of more which-- the ones that would add after some testing would be some PostgreSQL to manage, basically PostgreSQL's clean up, like the vacuum commands, and clearing session log commands, et cetera. Some maintenance stuff that really could be added to this, but right now there's just four in use. We use certbot for the SSL certificate. So something to regularly renew that.

Server reboot, which I added that a long time ago after I had found some Apache-- there was some Apache crash that would happen and a reboot would fix it. It just reboots in the middle of the night. And like I mentioned, we have a backup and daily backup that runs and then pushes the backup file to S3. Those are just PG dump tar files from the database. And then I've started to build out some email reporting, so I made this one email that sends weekly, that says how many new reports have been made. Plenty more work to do. That could really make that a lot better.

And so now, here just to give a really quick overview of what I have in progress, or in store, so to speak. The land manager system has really ballooned since the initial creation of the project and the initial requirements. So I'm going to be refactoring that and essentially my plan is to build out a new user profile that has some more attributes in it. And so I can push a lot more standardized information into one user profile, and get away from using groups, the Django groups, which is not really the best way to handle some of this. But merge that profile into the current user profile page.

I'm also looking at redoing, or adding geo-permissions. The idea is to be able to add Django features which are the park boundaries, so to speak, or any other unit boundaries. And then attach those boundaries to the user profile. And run an additional set of filters on the boundaries that are attached to the user requesting the page. And one of the biggest reasons for this is that the staff at FPAN are very comfortable using the Django admin. So this would give them better control over adding new users, and then assigning those users land management units or groups of units.



And then finally, it's been an issue for a while, the scout sign up that we created works fine, but it's not the best workflow. It's a little bit circuitous, so probably refactoring that either into the current Arches set up or at least just fixing it up a little more to make it better.

There's, like I mentioned earlier, plenty of other topics to talk about. I've done a lot of different pieces of this project to accommodate and build out different aspects of the entire thing. But I think I'm probably-- time is up. And wanted to get to questions, if anybody has any.

Hey Thanks Adam. That was great, I appreciate that.

Yeah, no problem.

I hope My internet was going in and out the whole time, that's pretty unfortunate.

No we just lost you for that one moment. So I do have a question that I'd like to pose, but I want to also let people know that we have about 15 or so minutes left in our meeting today. And I do want to open it up to a broader participation. So if you have a question, or a comment, or maybe a proposal, for what we should do on our next meeting, or if we should have a next meeting, or if so when should that happen? Please don't be shy. Feel free to unmute yourself, and jump into the conversation.

But while you're working up your courage, I'll ask Adam one quick thing which is this idea of using geometry to manage permissions to specific instances. It's something that we've been thinking about for a while as well. And it comes up pretty regularly from the user community. And I guess I'm curious to hear a little bit more about the approach you're taking. One of the things we're worried about is models with multiple geometry nodes, right? And how you would manage permissions based on nodes, and you can actually have multiple geometry nodes in a single branch. So have you been thinking about at that level, or are you still in the early stages of your thought?

Yeah, I've tested the things that I need to have happen, and basically they work. Like I mentioned, what really is difficult is applying, or I have found difficult, is applying these permissions to only one resource model. But yeah, ultimately, I'll just show the quick-- dimension to places that needs to happen, injecting PostgreSQL into the api.MVT class proved pretty easy. And so in this case, you can see this code, I'm pulling from this management area model which has a geometry attached, and then adding this where clause, which is just a PostgreGIS intersect based on the geometry in that model.

And then this just gets injected right in here. So this would, I think, handle what you're talking about where this is just pulling it down to the tile geometry, not the resources instance name. And it's also by node idea, of course. Which is which was my biggest concern in this case, because of-- well, it's not the biggest concern, it's one of them, because this is just driving the vector tiles. I was just working on this, this past week, it works fine. There's plenty more to do to, this is only proof of concept.

And then in terms of-- on the other side affecting the search results themselves, I was able to use basically the same logic to pull a feature geometry, and then push it into the search query. And this works. I pulled this basically straight from the map filter search, the Arches map filter. And so this is where I'm at with these things. And you can see it's pulling from a Django ORM object.

And so there's plenty, like I said, plenty more to do to really combine all-- to actually make this what I want, and also to apply this to only one graph ideas. Note it's obnoxiously difficult. But anyways, yeah, I hope that answers your question. I don't know if this is filling all of the spaces you've been talking about. But again, this is coming from, OK, we've been expanding the conditions for permissions for a long time, and this is my new idea for fixing up some of the long term patches that I've had to make.

Well, thanks. Yeah, that's really illustrative. I appreciate it. I'm just looking at our chat window here, and there is a new question from David Myers at the Getty, about the Arches Esri integration, and he's asking if we can provide a bit some information about where things stand with that, and I'll be happy to start that. And Andy if you want to jump in when I'm done, please feel free to do so.

For those of you who are unaware of what we're talking about here, we've been working on a real time integration between Arches and Esri's desktop GIS product, specifically ArcGIS Pro. And we've built an add-in that you can install an ArcGIS Pro, that lets you access Arches data directly from within the Esri product. So you can pull in Arches model instances as feature classes, native feature classes in Esri.

And you can create new geometries and new instances inside of ArcGIS Pro, and then upload them to Arches. And then in fact, open up the Arches user interface within the Esri product. And take advantage of the Arches UI to build out all the rich detail of a new instance. So that's what we're doing in a nutshell. Hopefully, that gives you a sense of what we're trying to accomplish.

It's really making it possible to create and update instances, specifically the geometries of instances in Arches, but using native Esri tools. And have that work really in a very seamless way. And I guess, the state of the work is that that's really running at the moment. It seems functional, there are a number of folks at Historic England who have been doing testing with it. Who've identified a relatively small number of little bugs and technical issues that we're currently looking at how to resolve. But the bottom line is we expect that this add-in will be available for just general use in the near term. It is on GitHub, the base is publicly available. So there's nothing stopping you from looking at it and jumping in and testing it if you want to.

A couple of caveats, it does require Arches version 5, 5 at least-- maybe even 5.1, I'm not 100% sure. I can't remember off the top of my head. And it requires ArcGIS Pro. It has a new dependency something called COOP, which is an open source product that was written by Esri. That lets you take JSON endpoints and process them into native Esri feature services. So that's a new requirement to support this. That's also a publicly available repo on GitHub. So if you're at all interested in this, give us a shot we can get you started in the right direction. Andy, do you have anything to add to that.

No really, I think that's pretty comprehensive. I think just to clarify that the COOP element you guys had working up for version 5.0 but the add-in is for 5.1.

Right.

But everything you said is fine here, we're just assisting with the testing. Just to help you guys along.

And one thing I would point out here is, it's a good example of how much better a product becomes when we work together. One of the things we did early on was-- here from Historic England-- what the requirements were for this, the add-in Esri group looked at, how to implement this, we looked at how to implement it, I think we came up with a pretty good solution testing this-- one of the things that fell out of it was a particular network configuration inside HE, which Andy was good enough to look at, and resolve and provided a fix to the code to deal with some network configuration issues within HE in a very generic way.

To me that's a really great example of why we do it the way we do it, right? Which is look for specific use cases, and the opportunity to partner with people who are implementing Arches, so that not only we'll write the code, but we can deploy it and test it in a way that is reasonable. And gets to-- actually the core thing which is something that works generically, but also within a specific constraints of-- like in this case HE.

I do want to make sure that we open up the conversation. If you have any questions about what you saw today, again, please don't hesitate to chime in. But one of the things I do want to ask the group is whether this was a useful use of Time I mean, I realize that it's a bit challenging to get everybody to sit down in front of their computers at the same time, given the time zones that separate us.

It'd be great to hear from you whether this second Arches developer meeting was worth it for you? And if it was, would it make sense to do a third meeting? And if so, what should we talk about in the third meeting? Would it be best to continue on in this vein where participants share the work that they're doing? Would it make more sense to have meetings that are more interactive where we talk about deployments, and/or code, or specific topics and code. I'm open to any of those things, but hearing from you all what you think might be most useful would certainly be helpful.

I think-- sorry, Dennis. It's Andy, one of the things, I think, we had a few years ago was we all got together in Swindon in the UK. And had a training session, which some of your guys-- Robert Cyrus came over and took us through some really interesting stuff. And I'm wondering whether maybe a future event could potentially have something less along the lines of this is what we're doing, but maybe more along the lines of this is an example of some of the stuff you could do, and maybe a run through of developing some of the extensions you can do in Arches.

Yeah, I agree that would be really cool. And meeting in person lends itself to that. I think maybe more so than a Zoom call. But when the world allows us to get back together again in the same

place at the same time, I'd be super up for that. We originally had the idea of hosting something like that in San Francisco. And that it looked like there was interest in that.

Of course, there's some challenges with that, I mean getting the money and the time set aside to actually make the trek to a place like San Francisco, or London, or New York, or Louisiana or wherever we'd end up doing it. Does, I think, somewhat limit the participation. But I think it's really valuable to be able to sit down and share code, share ideas, talk about what's possible with Arches, and then actually sit down and write a bit of code to explore whether ideas could actually be implemented.

So I'm done with that, for sure it would be great to be able to do that. And if there's a way to do it remotely before we can all be in the same place at the same time, I'm pretty open to it. I mean, Andy, do you have some ideas about how we might do that in the near term?

No. I mean, maybe not a webinar. This is quite useful to see what everyone is doing, really, to get a flavor for it. Maybe as a separate exercise, perhaps we could run things maybe a community-- maybe recorded stuff, because obviously sometimes this is not particularly useful to have everyone come together live to watch someone do some coding. Perhaps maybe it's actually something that could be done separately, rather than the regular meet ups. But until there's a vaccine, I think we're going to be struggling to meet up in person.

Yeah, I think that's right. I do think a smaller group of people could probably do what you suggest. Which is have a conversation with sharing screens and writing code in a collaborative way to explore whether a particular set of new functions could be implemented or test new ideas. That would be possible, and there's certainly nothing stopping groups of people doing that. It would be great if that does happen, to do a recording and make that publicly available, or somehow summarize what was done and share that back with the community. I mean that would be really, really good and I think super useful.

I mean it goes to a bigger point here, which is I'm actually really very, very excited with the level of interest there is in Arches. And the number of really interesting implementations that are ongoing. I mean, we saw three today that were really, really cool. And it feels like there's a bit of momentum starting to build up around developers who are doing interesting things with Arches.

And I know there are others that didn't get a chance to present today. I guess, I would just be really in favor of having people post back to the community the work that they're doing, or the little technical breakthroughs that they've made. Just as a way of starting conversations amongst yourselves, because I think that is one of the key things about a healthy developer community, is the willingness to share and explore together. And I feel like we're really close to that.

Dennis, this is Adam. I don't know, I'm preparing for this presentation today. I feel like I could do 40 more. And I think everyone probably could, because there's just so many different pieces to talk about. And so many different, of course, different installations but by different people. And people do things differently, of course. So part of me likes the idea of doing this exact same thing in a few months, and just finding three more people. And maybe that sounds basic, but this

worked out pretty well. And I just think there's so many more people to hear from. It would be a specific topic, or it could be here's this installation not that I did so, I guess, I'm just saying, I'm all for more. And I think there's plenty more to talk about.

Yeah, I agree, Adam. I made a decision to not show up any of the stuff that we're doing, because I felt like there was already enough being done that it would have been too much in one meeting. But I think it would be easy to do this again in say, late winter, or early spring next year. So like Marchish time frame. And I think we'd have little trouble in finding three more projects to highlight and talk about. And I'm happy if that's the case.

It'd be good to hear, again, more broadly from the folks on the call today, whether that would be something that would be considered useful and helpful. And I'm sensitive to the time that we've got, I mean, it's been two hours. And I know from personal experience how hard it is to sit through a two hour web meeting. So I don't want to drag this on.

But what I will say is a thing that you could do would be to get on the developer forum, and provide some feedback about what went well today, what didn't go as well as it could have gone, and what you'd like to see in the next available meeting. And it doesn't have to be a book. It could literally be a couple of bullets, like, hey, today's meeting was fine. And the next meeting will be great to have three more presentations. That'd be a perfectly useful and helpful piece of feedback.

Equally helpful would be, today was a good try. Here's what didn't work as well as it might have, here's some suggestions for what you might think of the next meeting. And we'll take those just as seriously as the, hey, that was a good job. So if you do have an opinion, please go to the forum, and let us know if this was helpful or not. And also give us a sense as to whether the time frame for the next meeting makes sense. I will I'll just throw out that March seems like a reasonable time to consider the next meeting. If that seems too early, too soon, or too long of a wait, again, just give us a sense of that in the development forum.

So I appreciate that, I'm sure the rest of the speakers will as well. With that, I'm going to invite any last comments. If you have something that you're really burning to say, please feel free to unmute yourself. And if we don't hear anymore, let's just maybe put a bow around this today. I thought it was really great. Kudos to all the speakers. I felt like each of the presentations was super interesting. And it was really, really inspiring to see the work that's being done on the platform. I can just-- I know, I can speak with the Farallon team in saying that, seeing the work that you're putting into the platform, and the investments you're making in both time and resources, it just speaks volumes to the good work that you're doing. So I really appreciate that.

And I'll say once again, thanks everyone, for participating. And we'll talk to you again in 2021.

Thanks Dennis. Thanks guys. Thanks. Thanks.