Arches Implementation Considerations

The following document seeks to help an organization that is planning to deploy Arches. All considerations are highly variable, but the guidelines below should help.

DEPLOYMENT CONSIDERATIONS

Arches will need to reside on a server, either on an in-house server or on a cloud hosting service, or perhaps even a combination of the two. This may not be the type of software deployment that you are used to, so work your way through the following considerations to get a better idea of where and how to host Arches:

Institutional hosting requirements and rules
Begin by figuring out if your institution has rules you'll need to follow when it comes to hosting databases and websites. In some cases, an in-house server is the only option, and in others, no such option will exist. This is important to consider up front, as it will impact the overall cost of the project.

Technical specifications
The Arches documentation recommends at least 4gb of RAM for evaluation and testing (8-16gb for production) and 10gb minimum disk space to install the code base and test dataset. However, required disk space really depends on the size and type of the data you'll be storing. Do you have a lot of photos or videos? These types of resources will use much more disk space than a simple historic resource.

Which operating system (OS) will work best for Arches?
Arches was developed in a Linux environment, but is designed to work with Windows as well. If you have full control over your choice of OS, Linux (Ubuntu) would probably be the easiest way to go, as Windows installations require a few extra steps. Other flavors of Linux (RedHat, CentOS) are supported as well.

Which cloud hosting service will work best for Arches?
If you are going to use a cloud hosting service, there are many good options—Google, Microsoft and Amazon all offer cloud hosting services, as well as smaller companies like DigitalOcean. Any of these will work for Arches. However, at this point, the most extensive tests and deployments have been done using Amazon Web Services (AWS). We have found AWS to be very effective in that it is flexible, well-documented, and cost-effective.

Ongoing maintenance and server administration
It will be necessary to establish a system administrator, potentially train that administrator as needed, and determine who will provide ongoing technical support. Ongoing support will consist of basic server updates, as well as Arches upgrades or enhancements.

DATA CONSIDERATIONS

An Arches database can hold data of all types and formats, so there is very likely to be some legacy data migration involved in any Arches installation. Use the following considerations to think about how your database will handle data:

Legacy data types

Arches can hold spatial data, tabular data, and any kind of digital file. What kinds of data will you be migrating to the new database, and how much of it? What kinds of file formats do you have? The first step is to make a thorough collection of all the legacy data you wish to migrate, and understand all of the data fields that exist in each dataset.

Resource graphs

The structure of your Arches database will be defined by resource graphs. Therefore, the first consideration is how well your legacy (and future!) data will fit the default Arches-HIP resource graphs, which you can find on the Arches website (www.archesproject.org). You'll be able to modify these resource graphs if you like (see Database Customization below), but you'll want a big picture understanding of the whole database before you begin to do so. Remember, migrating to a new database system is the best time to change old procedures!

Migrating data

Arches allows legacy data integration by the use of a single .arches file, so you'll need to create a conversion process to move from your various data formats into this file format. The .arches file format is very simple, so any existing format—MS Excel, ESRI .shp, Access databases, etc.—can be converted, if you can set up a good process to do so. Converting your data will be an involved process, and the most difficult part will be mapping your existing data fields to the Arches resource graphs. That's why it's imperative to begin by inspecting your legacy data, and defining your resource graphs.

SOFTWARE CONSIDERATIONS

An organization implementing Arches must understand the level of effort and technical skills necessary for the installation process. Though all installations will require a **basic configuration**, further **database customization** may be necessary, or you may be interested in **extending Arches** with additional components. We've tried to lay these out as different "levels of effort" below, to help in the planning process.

Basic Configuration

An Arches installation may only require this most basic level of configuration. If this is the case, you'll only need to make changes to your settings, content, and controlled vocabularies.

Localized Settings & Content

There are a number of settings and a bit of content that will need to be localized. For example: default map coordinates, home page content, your organization's logo, additional overlays, configuring your basemaps (or get your own Bing key to continue using theirs), etc. You may also want a non-English language for your installation's UI—Arches uses Transifex to manage its ever-growing number of translations, so you can set your deployment to use one or more languages for which a translation is available. *Changes to the settings or content can be made at any time during the installation process (or even after the database is in production).* Skills: HTML, CSS, basic text editing

Controlled Vocabularies
Though Arches comes with a predefined set of controlled vocabularies (which are stored in the authority documents) you should expect to update these to reflect your local terminology. *Changes to controlled vocabularies can be made in the RDM after you've installed the database, but because they are initially loaded directly from the authority files, it's recommended that you edit the authority files before your final install.*
Skills: Basic text editing (or MS Excel), experience with controlled vocabularies

Database Customization
After working through the Data Considerations listed above, you may determine that changes (or additions) to the default Arches-HIP resource graphs are necessary. This will require work on the back-end (to redefine the resource graphs) as well as on the front-end (to support these changes in the data entry forms and reports). Thus, database customization should be considered a two-part effort:

Resource Graphs
First, you must modify or create the updated resource graphs. We use Gephi to do this, but there are other open source graph visualization software alternatives as well. Start with the default resource graphs for working examples. *Changes to the resource graphs will require a database reinstall, and possibly new controlled vocabularies, so this effort should be made at the beginning of the installation process.*
Skills: Gephi (optional), data modeling/experience with controlled vocabularies

Data Entry Forms & Resource Reports
Second, change the HTML templates, JavaScript files, and Django models that are used to create the data entry forms and resource reports. *Changes to the forms and reports can happen at any point in the installation process, as long as the resource graphs have been defined.*
Skills: HTML, CSS, JavaScript (bootstrap, knockout), Python

Extending Arches
The Arches platform is open and malleable, so your imagination is really the limit of how your installation can be extended to support a rich database and unique user experience. For example:

- Do you want…
  - …the login page to be the first thing that a user will encounter?
  - …the ability to show historic maps as overlays?
  - …to incorporate 3D model viewing into resource reports?
  - …to implement a draft → publish workflow for resource creation?
  - …to add the ability to geocode addresses to create spatial coordinates?

Customizations such as these can be seamlessly integrated into your Arches app code. We only ask that you share your improvements with the rest of us!

Skills: Depending on your objectives, HTML, CSS, JavaScript, SQL, and Python may be necessary for this step. Specifically, familiarity with the Django web framework, OpenLayers 3 web-mapping library, PostGIS databases, and ElasticSearch are likely to be very useful skills.