Software for Digital Preservation

- This section will introduce Open Source software and how it can be used for digital preservation.
- This will include the history and ethos of OSS, the pros and cons of using this type of software and information on how to get started using OSS.

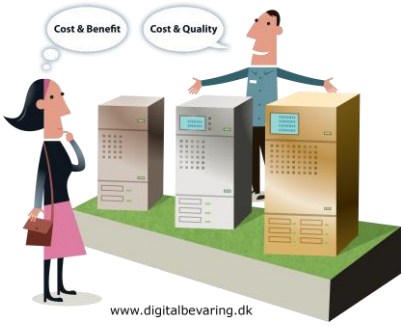
Software for Digital Preservation

- Two main types of software for digital preservation
- Large-scale applications:
  - Repository systems
  - Storage
  - Workflow
- Tools for particular functions:
  - Characterisation
  - Migration
  - De-duplication
  - …

- When looking at OSS software for digital preservation there are two main types of product you may consider using.
- The first are large scale applications which can be used to manage multiple processes.
- These can include complete repository systems, software for managing storage and workflow management systems for implementing potentially complex process.
- The other main type of OSS for digital preservation is smaller tools that carry out particular functions, which can be smaller-scale processes or a step in larger processes.
- These can include tools for characterising a digital collection, for migrating a particular file type or to check a folder for duplicate file.
- There are many of these smaller tools and often several that will carry out the same function.

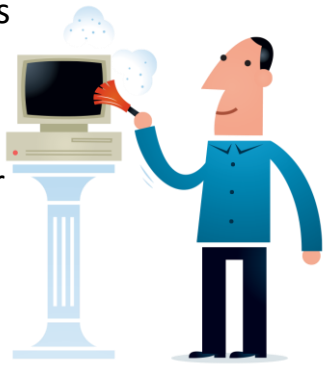
A Key Decision….

Vendor or Open Source?



History of OSS

- First conceived in late 1990s
- Adopt best practices from Free and Commercial Software
- Open development = better software
- First program released as OSS: Netscape browser
- Server/software infrastructure early priorities



- The concept of Open Source Software (OSS) was first introduced in the late 1990s as an evolution of the Free Software movement.
- It was created with the idea of adopting the best practices from both Free and Commercial software development.
- They hoped to retain the superior open development model of Free Software which had been proven to produce better software.
- This would be couched it in a more structured (but open) legal framework.
- The first program to be released as OSS was Netscape's browser, the code for which has since become the basis for the development of several other OS browsers including Mozilla's Firefox.
- Early efforts in the OSS domain focused mostly on server and software infrastructure projects but has since expanded to include all forms of software.

A Free Beer, A Free Cat, or Free Speech?

A Free Beer

- OSS is not necessarily free as in 'gratis'

A Free Cat

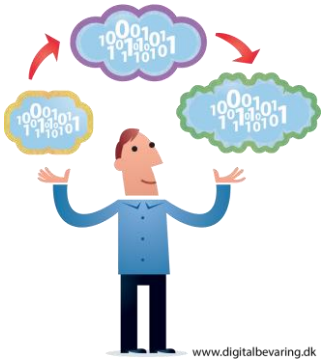
- Costs relating to implementation, upkeep, training, support, etc.

Free Speech

- Access to source code
- Ability to adapt to own needs
- Can redistribute

- Freedom and openness are key to OSS but many mistake this to mean that the software should be available free of charge.
- This is not the case and the freedom of OSS is often expressed using the analogies of 'a free beer', 'a free cat' and 'free speech'.
- If you receive a free beer, this is something that comes to you at no cost and you can consume without any further ramifications other than slight inebriation.
    - This is not the type 'free' that applies to OSS. It is often offered for no or a low cost but there is no requirement to be free as in 'gratis'.
- Some have likened OSS instead to the idea of a free cat; while the original gift may not cost you anything, caring for the cat will cost money for food, toys, vet bills etc.
    - With OSS although the original software may be free or relatively cheap, you will likely incur costs relating to implementation, upkeep, training, support and other issues.
- The other essential freedom of OSS has been likened to free speech in that there is a requirement for free access to the source code, to adapt the software if desired and to freely redistribute the product.

Development Model



- Users as co-developers
- Early releases
- Frequent integration
- Different versions: beta vs stable
- High modularization
- Dynamic decision-making

- There are several key ways in which the development of OSS differs from commercial solutions, these all aimed at creating more complete and stable products.
- The differences include:
    - Users are considered to be co-developers alongside programmers. This emphasises both the collaborative nature of OSS as well as the belief that testing and bug identification are as important to the development process as writing code.
    - Programmers are encouraged to release code as early as possible to allow the interaction described in the previous point, users can check functionality is fit for purpose and spot bugs early. This input leads to more productive development cycles.
    - Multiple programmers may be working independently on the product so they are encouraged to frequently integrate their work to ensure consistency.
    - The creation of modularized products which make it easier for multiple people to work on the product as well as enabling customization and updates.
    - Dynamic-decision-making is encouraged to ensure changes can be

incorporated quickly.

Different Types of Contributions

"Give as you can"

Help with:

- Scoping developments
- Identifying requirements
- Writing code
- Providing feedback
- Identifying Bugs

- As mentioned in the previous slide, the term 'developers' is used quite widely in the OSS world, including more than just those creating code.
- This is an important factor to remember if you are planning to use OSS but do not have the skills or resources to contribute to the programming efforts.
- Contributions are encouraged on a "give as you can" basis and all types are equally valued.
- These can include helping:
    - Make suggestions for and scope new developments
    - Identifying the more details requirements for developments
    - Contribute to the writing of code
    - Providing feedback on new functionality to make sure it is fit for purpose
    - Identifying bugs early to create more stable software
- Even a small amount of time spent on one of these activities helps the community at large.

OSS Licenses

- ‘Copyleft’ licenses
- Approved by OSI
- Emphasis on collaboration, openness and reuse
- Derived works must have same license
- Popular licenses include:
  - Apache License 2.0
  - GNU General Public or Library General Public Licenses
  - BSD 3-Clause or 2-Clause Licenses
  - Mozilla Public License

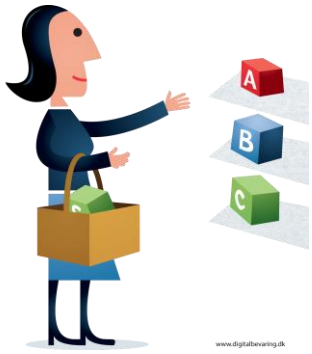
- The types of license used with OSS are often referred to as ‘copyleft’ as their emphasis is on providing a framework for freedom of use rather than focusing primarily on restrictions.
- To be accepted as a true Open Source license it must be approved by the Open Source Initiative. They maintain a list of approved licenses on their site, which also includes information on the most commonly used.
- OSS licenses are written to encourage collaboration, openness and reuse, with proper attribution to the original creators one of the few restrictions on reuse and redistribution. They are usually far simpler than their commercial cousins and a fraction of the length.
- They also require that all derived works adhere to the same license. Ensuring the ethos of OSS is passed to those works.
- Although there a large number of custom OSS licenses, many projects choose to use one of the more common standard licenses listed here. More information on these can be found on the Open Source Initiative website.

GitHub

- A code hosting platform
  - Collaboration
  - Version Control (Git)
- Used by developers of the majority of OSS digital preservation tools and solutions
- Public and private development spaces
  - Basic account = free
- Access to full source code
- A way to contribute

- GitHub is a platform for hosting software source code.
- It allows developers to collaborate on the creation of software no matter their location and to managed version control through GitHub's Git solution.
- GitHub is the most popular online code hosting platform and is used by the majority of digital preservation-related OSS development.
- GitHub provides spaces for both public and private developments and basic accounts, which allow participation in public projects, are free.
- A project's 'respository' will provide full access to the software's source code. The repository is the name used by GitHub for a particular project.
- Interacting with developers on GitHub is the best way to help identify bugs and make suggestions for new functionality.

Things to Consider When Selecting OSS

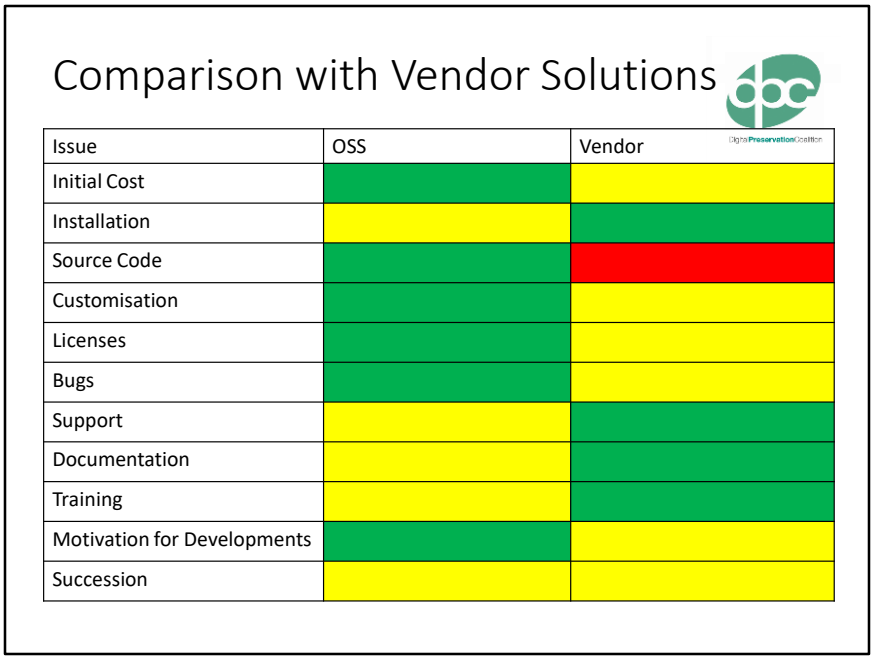


- Longevity
- Stability
- Costs
- Ubiquity
- Skills required
- Documentation/training
- Compatibility

There are lots of factors to consider when choosing an OSS solution or tool, but a short checklist of issues might include:

- **Longevity** – How long has the software been available? Does it have a robust and active community of support?
- **Stability** – Do user comments indicate that the software buggy? Has a stable version been introduced?
- **Costs** – Is there a purchase cost? What will be the costs for implementation? Will you need to pay for support?
- **Ubiquity** – Is the software used by similar organisations? Can you rely on peer to peer support?
- **Skills required** – Do you have the necessary skills required to implement and use the software? If not, would they be easy to acquire?
- **Documentation/training** – Is the software supported by good documentation? Are there training resources available?
- **Compatibility** – Is the software compatible with your systems and other solutions or tools you have or would like to implement?

Comparison with Vendor Solutions



Issue	OSS	Vendor
Initial Cost		
Installation		
Source Code		
Customisation		
Licenses		
Bugs		
Support		
Documentation		
Training		
Motivation for Developments		
Succession		

- The table on this slide shows a simplified visual comparison of using Open Source Software versus vendor provided solutions. Green = good in this area, yellow = mixed, some strengths and weaknesses, red = not available.
- Both have their strengths and weaknesses. Looking at these in a little more detail:
    - **Initial Cost** - Much OSS is free but even if there is an initial cost in procuring OSS it likely to be small in comparison with vendor solutions which may require a significant investment as well as an ongoing commitment to additional services or updates.
    - **Installation** – Vendors are likely to provide support with the installation of more complex pieces of software and simpler pieces are distributed in an executable format that is usually easy to install. Installation of OSS is more variable and may require an invest of resources and more technical skills to get it up an running.
    - **Source Code** – OSS provides access to the original source code whereas vendor supplied solutions are pre-compiled.
    - **Customisation** – By having access to the original source code, this means it is possible to fully customise OSS for your organization if you have sufficient programming skills. There is also the potential for greater customisation and collaboration from the wider community. Customization of vendor

solutions is usually limited to the in-programme options and tools. Any more significant customisation will depend on the vendor and their priorities. Some vendors will create customised modules for their software for a fee.

- **Licenses** – It is generally only to acquire only one license for OSS no matter how many installations are needed. They are also more open to the creation of derivative works and redistribution. Vendor licenses tend to be far more restrictive, limiting how and where the software can be used. It also normal that multiple licenses may need to be purchased one for each user or installation of the software.
- **Bugs** – Due to the collaborative nature of OSS development stable versions of the software tend to be less buggy and when bugs are identified they are addressed more promptly if a reasonable-sized community exists for the product. Vendor software tends to be more buggy when first released as getting the product on the market is a key. They may also be slower to address identified bugs later depending on their current commercial and/or development priorities.
- **Support** – Vendor software often comes with a support package, or this can be purchased as an addition. Meaning that there is some expectation of good and prompt support. The situation is more mixed for OSS software and depends on factors such as the size and engagement of the user community or the availability of paid-for support services.
- **Documentation** – Documentation was historically poor for OSS but the situation is much improved in recent years but it cannot be relied upon for all software. For vendor solutions, there is a reasonable expectation that good documentation will be provided when the product is procured.
- **Training** – Like documentation, training for OSS is mixed and sometimes only available for larger/more established programmes. Commercial vendors will more likely have training resources available. Depending on the size and complexity of the software this may anything from online resources to the provision of in-house training for staff.
- **Motivation for Developments** – One of the key strengths of OSS is that developments are normally motivated directly by the needs of the user community. The main motivations for vendors are usually focused on commercial concerns; such as making a profit and strengthening their position in the market. This can mean they are less responsive to user needs and will adhere to business models such as planned obsolescence.
- **Succession** – No matter the type of solution chosen it is important to carry out succession planning to make sure data can be retrieved in the event of the discontinuation of the solution. With OSS the continued support and development of a product relies on the ongoing engagement of the community. If this abruptly ends, access to the source code means users

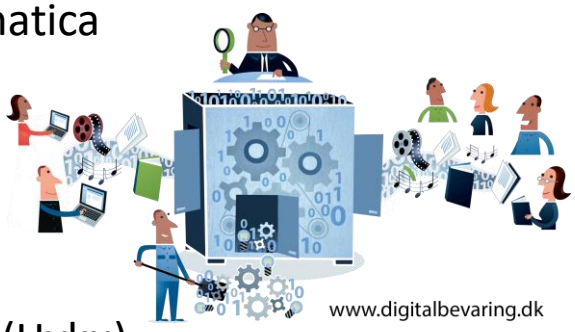
are in a strong position as long as they have the skills/resources required. With vendors, it is very important to include succession planning in any service agreements but issues may still occur in cases on bankruptcy.

Vendors/Service Providers

- Arkivum
- Artefactual
- Ex Libris
- Hanzo
- IMR
- Keep Solutions
- Libnova
- Mirrorweb
- Preservica

Open Source Repository Systems

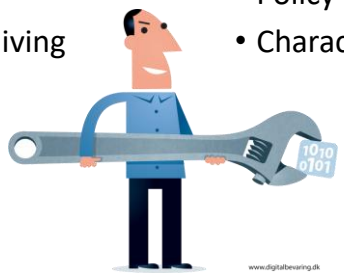
- Archivematica
- RODA
- DSpace
- Fedora
- Eprints
- Samvera (Hyku)



- If you wish to implement a full repository system for your digital collections, there are an increasing number available.
- Archivematica and RODA are two examples of repositories that are well supported by both a specific organisation and their user community. Both system offer free repository solutions with additional plug-ins and paid-for support services.
- DSpace, Fedora and Eprints have emerged from and generally been used more in the research data and publication domains but have been implemented by a variety of different organisations.
- Samvera is a repository solution that has evolved from the Hydra project and collaborative work of a number of Higher Education institutions. Their Hyku solution aims to be an easy to install ‘out of the box’ repository.

Types of Tools

- De-duplication
- Forensics
- Decryption
- Fixity
- Web Archiving
- Migration
- Emulation
- Validation
- Policy and Planning
- Characterisation…..



- There are many other types of OSS tools for digital preservation such as tools for:
  - Identifying duplicate files
  - Carrying out forensic analysis of files or directories (particularly useful for handling processes such as disk imaging as well working with protected or sensitive files)
  - Decrypting files
  - Checking file integrity using fixity values
  - Carrying out preservation planning
  - Migrating file formats
  - Accessing file using an emulator
  - Validating a file format matches the format specification
  - Helping to write policy
  - And many other tasks and processes….

Example Tools: Characterisation

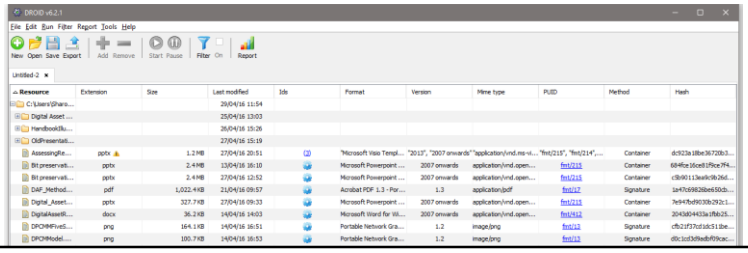
Various tools with different functionality:

- DROID
- Apache Tika
- C3PO
- FIDO
- JHOVE
- FITS

- One of the most widely used types of OSS tools for digital preservation are those used for characterisation, and several are available.
- DROID is developed by The National Archives of the United Kingdom and links to their PRONOM database of file format information. It is one of the most widely used characterisation tools as it is available with a graphical user interface and includes functionality such as fixity checking.
- Apache Tika, C3PO, FIDO and JHOVE all offer similar functionality but with different strengths and weaknesses. For example, JHOVE provides the richest output including file format validity but only for a limited number of format types.
- FITS (the File Information Tool Set) is a little different from the other tools as it actually packages together a number of the other characterisation tools including DROID, Apache Tika and JHOVE. This means it can produce rich results but also inconsistencies between the different tools.

Basic Characterisation: DROID

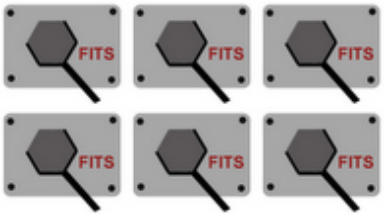
- Works with PRONOM file format registry
- Analyses contents of folder(s)
- Captures information such as:
  - File name, location, file size, last edited, format, version, PRONOM ID, checksum
- Outputs raw data or a variety of reports



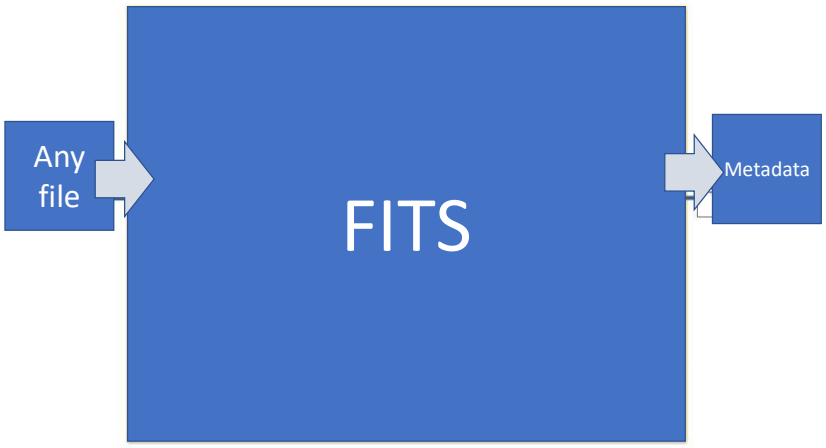
Delving Deeper: FITS

http://projects.iq.harvard.edu/fits

- Wraps together a selection of open-source tools
- Identifies, validates and extracts technical metadata
- Command line operation
- Consolidates info
  into an XML file



FITS – File Information Tool Set



A Brief Intro to XML

"a **mark-up language** that defines a set of rules for **encoding documents** in a format that is both **human-readable** and **machine-readable**…."

```
<book>
        <title>Cold Comfort Farm</title>
        <author>Stella Gibbons</author>
        <publication edition="1st">
                <date>8 September 1932</date>
                <publisher>Longmans</publisher>
        </publication>
</book>
```

Attributes

FITS Output – File Info

<fileinfo>

<size toolname="Jhove" toolversion="1.5">6406168</size>

<creatingApplicationName toolname="Jhove" toolversion="1.5" status="CONFLICT">Adobe PDF Library 10.0.1; modified using iTextÃ‚Â® 5.3.1 Ã‚Â©2000-2012 1T3XT BVBA (AGPL-version)/Adobe InDesign CS6 (Windows)</creatingApplicationName>

<creatingApplicationName toolname="NLNZ Metadata Extractor" toolversion="3.4GA" status="CONFLICT">Adobe PDF Library 10.0.1; modified using iText 5.3.1 2000-2012 1T3XT BVBA (AGPL-version)/Adobe InDesign CS6 (Windows)</creatingApplicationName>

<creatingApplicationName toolname="Tika" toolversion="1.3" status="CONFLICT">Adobe PDF Library 10.0.1; modified using iTextÂ® 5.3.1 Â©2000-2012 1T3XT BVBA (AGPL-version)/Adobe InDesign CS6 (Windows)</creatingApplicationName>

<lastmodified toolname="Exiftool" toolversion="9.13" status="CONFLICT">2015:09:17 09:03:16+01:00</lastmodified>

<lastmodified toolname="Tika" toolversion="1.3" status="CONFLICT">2014-09-19T13:06:41Z</lastmodified>

<created toolname="Exiftool" toolversion="9.13" status="SINGLE_RESULT">2013:12:04 17:25:56+05:30</created>

<filepath toolname="OIS File Information" toolversion="0.2" status="SINGLE_RESULT">D:\Apps\fits-0.8.4\LargeScaleDataAnalytics_eBook.pdf</filepath>

<filename toolname="OIS File Information" toolversion="0.2" status="SINGLE_RESULT">LargeScaleDataAnalytics_eBook.pdf</filename>

<md5checksum toolname="OIS File Information" toolversion="0.2" status="SINGLE_RESULT">6e3d47cfdd7010adb6f0ffede28db303</md5checksum>

<fslastmodified toolname="OIS File Information" toolversion="0.2"

FITS Output – File Status

<filestatus>

<well-formed toolname="Jhove" toolversion="1.5" status="SINGLE_RESULT">true</well-formed>

<valid toolname="Jhove" toolversion="1.5" status="SINGLE_RESULT">false</valid>

<message toolname="Jhove" toolversion="1.5" status="SINGLE_RESULT">Too many fonts to report; some fonts omitted. Total fonts = 1118</message>

<message toolname="Jhove" toolversion="1.5" status="SINGLE_RESULT">Missing expected element in page number dictionary offset=1085132</message>

</filestatus>

FITS Output – Metadata

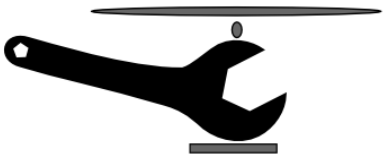

```
<metadata>
  <document>
   <title toolname="Jhove" toolversion="1.5" status="SINGLE_RESULT">Preface</title>
   <language toolname="Jhove" toolversion="1.5">EN</language>
   <pageCount toolname="Exiftool" toolversion="9.13">276</pageCount>
   <isTagged toolname="Jhove" toolversion="1.5">no</isTagged>
   <hasOutline toolname="Jhove" toolversion="1.5">yes</hasOutline>
   <hasAnnotations toolname="Jhove" toolversion="1.5" status="SINGLE_RESULT">no</hasAnnotations>
   <isRightsManaged toolname="Exiftool" toolversion="9.13" status="SINGLE_RESULT">no</isRightsManaged>
   <isProtected toolname="Exiftool" toolversion="9.13">no</isProtected>
   <hasForms toolname="NLNZ Metadata Extractor" toolversion="3.4GA" status="SINGLE_RESULT">no</hasForms>
   <standard>
    <docmd:document xmlns:docmd="http://www.fcla.edu/docmd">
     <docmd:PageCount>276</docmd:PageCount>
     <docmd:Language>EN</docmd:Language>
     <docmd:Features>hasOutline</docmd:Features>
    </docmd:document>
   </standard>
  </document>
 </metadata>
```

C3PO

- FITS: Content profiling tool
- Works with FITS to visualize FITS output metadata
- Setup can be a little more taxing

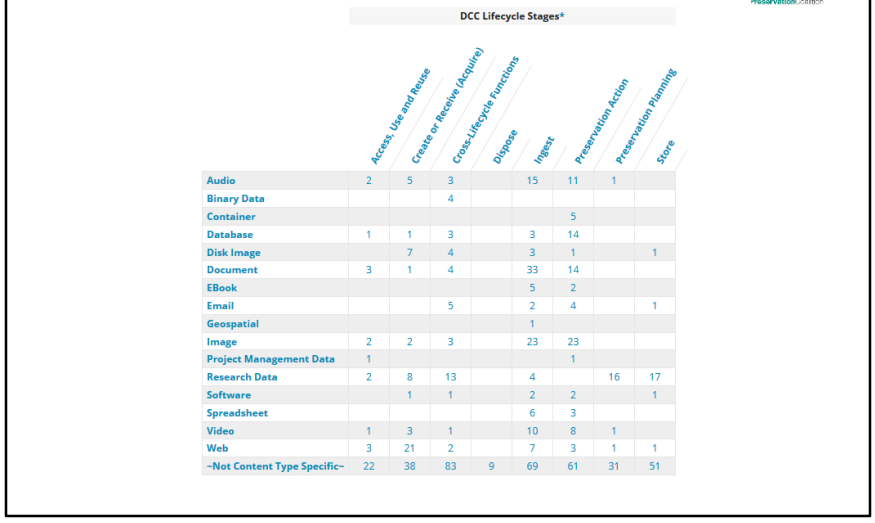
COPTR

- Tools registry for digital preservation
- Includes OSS and Vendor solutions
- Part of DigiPres Commons
- Hosted by the Open Preservation Foundation
- Browse by:
  - Name
  - Function
  - Type of content



- When looking for tools for digital preservation, one of the most useful places to start is the tools registry COPTR.
- The registry includes listings of both OSS and vendor software and solutions
- It is one of the information resources offered by the Digi Pres Commons and it hosted by the Open Preservation Foundation.
- COPTR allows you to browse tools by name, function and type of content.
- It is a community developed resource so the amount information varies by tool but contributions are welcomed.

POWRR Tool Grid



	DCC Lifecycle Stages*							
	Access, Use and Reuse	Create or Receive (Acquire)	Cross-Lifecycle Functions	Dispose	Ingest	Preservation Action	Preservation Planning	Store
Audio	2	5	3		15	11	1	
Binary Data			4					
Container						5		
Database	1	1	3		3	14		
Disk Image		7	4		3	1		1
Document	3	1	4		33	14		
EBook					5	2		
Email			5		2	4		1
Geospatial					1			
Image	2	2	3		23	23		
Project Management Data	1					1		
Research Data	2	8	13		4		16	17
Software		1	1		2	2		1
Spreadsheet					6	3		
Video	1	3	1		10	8	1	
Web	3	21	2		7	3	1	1
~Not Content Type Specific~	22	38	83	9	69	61	31	51

- The POWRR Project has been a major contributor to the COPTR repository and another eay to navigate the site is using the POWRR Tool Grid shown here.
- It allows users to identify relevant tools by object type and by lifecycle stages, which is particularly useful when developing new processes.