Adding Emulation Functionality to Existing Digital Preservation Infrastructure

Euan Cochrane
euan.cochrane@yale.edu
Yale University

Jonathan Tilbury
Preservica
Abingdon, United Kingdom
jonathan.tilbury@preservica.com

Oleg Stobbe
University of Freiburg
Freiburg i. B., Germany
oleg.stobbe@rz.uni-freiburg.de

ABSTRACT

The emulation of obsolete hardware and software environments to enable information to be read, and to facilitate interaction in a way that simulates the original user experience, is a well-established part of digital preservation solutions. Excellent tools have been developed to work with emulators, but these have remained in the research domain rather than being able to be exploited at scale. This paper explores a real example of how an emulation framework has been added to existing digital preservation infrastructure. This integration has enabled information has been extracted from obsolete hardware, held in a digital preservation system at Yale University Library (YUL), and linked to an appropriate emulator (provided by the University of Freiburg's Emulation as a Service framework). All of this enables YUL to recreate the user experience of interacting with content using the original software quickly and easily whenever a user requests it. This integration offers the prospect of large scale emulation as a service linked to real preserved data that is in need of this approach and this paper will examine the next steps needed to make this a reality.

KEYWORDS

Software-based Art; Emulation; Preservation Strategy

1 INTRODUCTION

An emulator is "hardware or software that enables one computer system (called the host) to behave like another computer system (called the guest)" [1]. Emulators enable users with newer computers to run software designed for older computers, software that may be incompatible with their current computer. In the context of digital archiving, preservation, and access, emulators have a number of useful applications. However, emulators have for a long time been difficult to configure and use, especially at scale, and require expertise that is rare amongst organizations that have taken responsibility for preserving digital content at scale. This (amongst other reasons) [1] has limited the acceptance of emulation as a viable long term digital preservation method. Fortunately in recent years tools that simplify the use of emulators in these contexts such as JMESS [2] and bwFLA Emulation as a Service (EaaS) [2], have become more readily available and are increasingly being used in production environments. With these emulation tools and services coming of age it is becoming increasingly realistic for digital preservation infrastructure providers and developers to consider how emulation can fit into their products and services. In this paper we describe how Emulation as a Service has already been connected with digital content stored in one digital preservation system Preservica, and discuss the work needed to further integrate and scale this approach for wider use.

2 CONNECTING EMULATION AS A SERVICE (EAAS) AND PRESERVICA: A CASE STUDY

Yale University recently implemented functionality to connect its new bwFLA Emulation as a Service (EaaS) implementation with its digital preservation system Preservica to initially provide access to content on CD-ROMs from Yale University Library's (YUL's) general collections. EaaS is a suite of software that simplifies the use of a variety of emulators enabling minimally trained archivists and librarians to make use of emulation technology in typical archival and library workflows. And it does so without requiring significant technical expertise or adding management burden to already busy work schedules by enabling access to preconfigured emulated computers via a web-browser interface. The EaaS software emulates a variety of different computer architectures behind the scenes so the power and value of the EaaS approach to the archivists or librarians it that it abstracts away the details of which emulator is being used and how it is configured, and simply provides the preconfigured emulated computers for use in archival/library workflows.

Preservica [2] is a digital preservation system that combines all the elements of the OAIS model [4] into a single system. It includes tools to ingest both simple and complex data objects, for example ISO disk images, and to store these in multiple data stores with full fixity checking. It also includes a flexible data management capability with an access module to allow the information to be searched, browsed and downloaded. At its core is a full file format preservation suite that identifies and characterises content and if applicable migrates it to new formats. It also includes a number of viewers to allow content to be rendered server side and delivered via a browser. This is built on a registry of file format information.

The initial connection implemented between Preservica and

[1] (n.d). "Emulator" in Wikipedia. Retrieved March 31st 2017 from https://en.wikipedia.org/w/index.php?title=Emulator&oldid=773101052 [2]"JMESS readme" in JMESS 2017. Retrieved March 31st 2017 from https://github.com/ jsmess/jsmess

[2] Preservica 2016. Retrieved March 31st 2017 from www.preservica.com [4]Technical Committee: ISO/TC 20/SC 13 Space data and information transfer systems. "ISO 14721:2012 Space data and information transfer systems – Open archival information system (OAIS) – Reference model", September 2012. Retrieved 31st March 2017
from https://www.iso.org/standard/57284.html

EaaS is a 'lightweight' integration using the standard Application Programming Interfaces (APIs) available in each product allowing access via an emulated computer by combining the following:

(1) An emulator
(2) A hard drive image with an operating system such as Microsoft Windows 95 installed on it
(3) (Possibly) some software for interacting with the content such as Adobe Acrobat Reader (4) The content itself

At Yale University Library (YUL) these assets are preserved within Yale University's local "Enterprise Edition" installation of Preservica. Within Preservica there are a number of collections that preserve the required assets:

(1) "Base-Images" collection
(2) "Software" collection
(3) "Derivatives" collection
(4) One or more content collections

The Base Images collection in Preservica contains hard drive images (single files that capture all the content on a hard drive) that have a minimal set of software installed on them such as an operating system. These, along with a configuration file, are all that is necessary to provide a basic emulated computer via one of the emulators available within EaaS [3].

Base Images are not provided as part of the bwFLA EaaS software suite and have to be curated by users of EaaS. At YUL the images were created by cloning or 'imaging' hard drives from original computers using digital forensics software (See Figure 1).

In most cases these images function without any changes within the appropriate emulator. In some cases, the images need to loaded in the emulator and have new hardware drivers installed in order to be compatible with the new emulated hardware.

Base Images could benefit from centralized solutions. Imaging original hardware to obtain base images creates an emulated computer environment is as close as possible to a functioning original machine ensuring a maximally authentic experience for the end-user in such a way that the user can seek out the original hardware to validate the experience of the emulated version against the original. However a wider implementation of this approach would not be sustainable even short-term due to a lack of available original hardware for each organization and long-term due to hardware degradation through natural causes. In the short/medium term this could be mitigated by establishing a network of hardware museums that could provide a valuable source for base-images and emulation validation services and in the longer term via thorough documentation of the original machines that will help to enable testing of future emulators. The preservation of the base-images will further ensure we have maximally authentic emulation experiences available.

To access Base-Images the EaaS service can be triggered to pull new content (using the Preservica REST API) from any pre-configured collections in Preservica and organize it into its administrative interface using buttons in the EaaS GUI.

Figure 1: Digital forensics software and hardware being used to capture the content of hard drives from original computers.

Users can then use the EaaS GUI to configure base environments. Once base images have been created or captured, and added to EaaS, staff can then add software applications to them. YUL's digital preservation policy framework [4][5] states that: "YUL will ensure access to hardware and software dependencies of digital objects and emulation or virtualization tools by [...] Preserving, or providing access to preserved software (applications and operating systems), and pre-configured software environments, for use in interacting with digital content that depends on them."

YUL has begun preserving software in compliance with this policy statement within a "Software" collection in Preservica. The "software" collection contains a set of installable software binaries used to install software. These in turn can be used to provide a maximally authentic rendering and/or interaction experience for researchers to access preserved digital content. An example might be the installation file for Adobe Acrobat Reader 4.05 from the year 2000. Currently the software installation binaries are all preserved in Preservica wrapped within disk image files (e.g. ISO files) to simplify their use with EaaS. However it is possible to submit files or folders of files to EaaS and have it automatically wrap them within a virtual CD or Hard Disk Drive (HDD) image and attach them to an emulated computer. The software binaries in the YUL collection are organized in a standardized structure with standardized metadata. To capture and arrange that metadata YUL is using a custom Metadata Encoding and Transmission Standard (METS) [7] implementation that references a persistent external master database, Wikidata.org[8], used to capture and store generic software documentation (such as title, publisher, input and output formats, etc). The METS file is also used to directly store localized information such as license keys, and local holdings information such as local media identifiers. In future iterations of this integration the additional software metadata necessary may be cached locally in Preservica's internal preservation metadata registry which already documents some relevant software applications.

[3] BIOS files and other cross-environment configuration files are considered part of the EaaS framework.

[4] Yale University Library, November, 2014. "Yale University Library's Digital Preservation Policy Framework". November 2014. Retrieved March 31st 2017 from http://web.library.yale.edu/sites/default/files/files/YUL%20Digital%20Preservation%

[5] Policy%20Framework%20V1%200.pdf

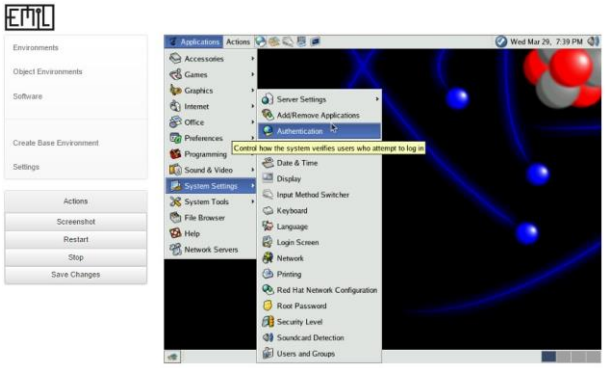

Figure 2: Configuring a base environment using the EaaS GUI

After ingesting a new binary or set of installation media into Preservica archivists or librarians can click in the EaaS GUI to "sync object environments" and import the software into the EaaS interface. They can then launch an emulated base environment and have the installation media for a selected piece of software automatically inserted into the emulated computer's virtual drive (either within a virtual CD/DVD drive or an emulated hard drive).

The staff member can now use the standard software installation process to install the software, shut down the emulated computer, and click a button in the GUI to save a new "software environment". The software environment is now ready to be exported for ingest back into Preservica for long term Preservation and future retrieval. This export and ingest process is currently manual however it is expected to be automated in coming months.

The software preservation workflow at YUL requires there is a basic record for the software in Wikidata, that the wikidata reference is in a METS file for the software, that it and the media images are ingested into Preservica, and that the software collection is synchronized with the EaaS GUI by clicking a button in the interface. Following this the software is installed and configured on a base-environment to create a "software environment". EaaS is then used to execute the software and document further metadata about it, such as the software's import and export formats (or 'open' and 'save-as' as the case may be) and the default format associations of the software as assigned in the operating system (in order to know which software will execute by default when a file of a particular format is opened). As of this article's publication date this metadata is being stored locally but in coming months it will be added to Wikidata to be shared with the wider digital preservation community.

Software environments can also be layered on top of one another enabling the creation of software environments that meet many different needs. As a worked example a user may configure a software environment that includes Microsoft Outlook in order to access the content stored in "PST" email archive files in their original context. On accessing the PST file they may discover that it contains a lot of .psd (Photoshop) files as attachments to the email. They can then go back to the Outlook software environment and layer Adobe Photoshop on top of it to create a new software environment that includes both Outlook and Photoshop. This will ensure that the Photoshop files attached to emails in the PST file can be accessed by users browsing the email records. Each of these shareable and reusable software environments can be exported and preserved in Preservica for preservation and future utilization.

YUL are initially using EaaS for providing access to content on CD-ROMs (CDs) that make up part of their general collections and date back as far as the late 1980s. The CDs contain a huge variety of content from government and business datasets, to interactive video and image content, to conference proceedings, and even computer games. The software requirements for accessing the content on these CDs are fairly minimal as the CDs were generally designed to come bundled with all software that was necessary to use them and simply require standard operating system base images along with some freeware such as Adobe Acrobat, VideoLAN Player (VLC), or Netscape Navigator.

3 ADDING CONTENT TO SOFTWARE ENVIRONMENTS

After configuring software environments in EaaS users can synchronize content from any collection in Preservica that EaaS has been configured to pull from, and which is structured according to a predefined (but configurable) substructure pattern within a Preservica collection allowing the EaaS automatically identify where the installation media are. METS metadata is also included to document information about which disc of a multi-volume set of media needs to be inserted into the drive first. This can be important when only one disc from a set includes the setup files or software, or when one disc is required to initiate an interactive experience such as a game or encyclopaedia, before switching to an additional disc to provide additional content. The documentation shows how the end-user GUI for EaaS can provide a button enabling users to change discs when interacting with the environment.

Once software environments have been configured and the content collections ingested into Preservica and synchronized to EaaS, content can then be associated with software environments to enable it to be automatically accessed by end-users in an appropriate software environment. Using the Emulation as a Service software organizations can choose between implementing three different workflows for matching emulatable software environments with digital objects in their collections:

(1) Automated
(2) Semi-automated
(3) Manual

Using the automated workflow the user selects an object to interact with via emulation. The content of the object is characterized, extracting file formats and creation dates of the files it contains, and matched to a database of available software using metadata held on each environment and an algorithm that attempts to find the environment containing the most compatible software. The results of this matching are either saved for a future request or are discarded and regenerated next time the object is requested. The latter approach may be advantageous for archives that are growing their software collection as at any point in time

[7]METS editorial board, 2010. "Metadata Encoding and Transmission Standard" April 2010. Retrieved March 31st 2017 from http://www.loc.gov/standards/mets/
[8]Thornton, K. and Cochrane, E. "Wikidata as a digital preservation knowledgebase" in The Open Preservation Foundation Blog. Retrieved March 31st 2017 from http://openpreservation.org/blog/2016/09/30/wikidata-as-a-digital-preservationknowledgebase/

the "best" environment for interacting with any particular object may have changed as new software was acquired.

Sometimes multiple matches may be relevant for a set of content, for example some CDs from the 1990s included both Apple Macintosh compatible content and Microsoft windows compatible content. In such cases the user can be given the choice of which environment to attempt to execute the content within.

With a semi-automated solution, the characterization and matching happen by default but a staff member confirms that the recommended environment is actually a good match for the files stored on the object by loading the environment and trying it. If there is an appropriate match the staff member can save the configuration as detailed below to be automatically provided by default to future users.

In the third, manual scenario, a staff member manually selects an environment for each object and configures it themselves. This is useful for complex environments or environments in which specialized software is provided with, or required by the object, or where custom environments need to be created for the content.

A core concept utilized in this implementation is that of 'derivative environments'. Derivative environments are complete software and/or content environments that are derived from and depend on a base-disk image that contains the full operating system installation and configuration files required to run the content contained in the derivative environment. The 'derivatives' collection in Preservica is where derivative environments, content files, and metadata files are preserved after being configured in EaaS. Derivative environments are created by staff in the processes outlined above whenever a software application or set of content is added to a base environment or software environment and a button is clicked in the EaaS GUI to save the environment as a derivative "object environment" (Fig. 6).

When saved as a derivative environment the EaaS software encapsulates the small set of data that was added to the base environment, or existing derivative environment, along with metadata instructing EaaS how to reintegrate the base with the derivative in real-time when requested in by users in the future. Periodically, derivatives can also be exported and re-ingested to Preservica for long term preservation.

4 ACHIEVING SCALABILITY

The use of EaaS with Preservica at Yale University is now relatively seamless and requires few advanced technical skills from the staff who will be continually configuring new derivative object environments as YUL adds more content to the system in the future. Such configuration is currently being undertaken by digital preservation staff but given the non-technical nature of the work involved, this task may be assigned to other library and archives staff in the future. For end users, the ability to click a link in a web browser and have an historic computing environment seamlessly load within the web browser is both minimally burdensome, and transformative from a research perspective. However, this ease of use for these users (be they researchers or non-preservation librarians and archivists) belies significant preparatory work on behalf of both the digital preservation team, the EaaS software developers and the community of metadata creators whose contributions behind the scenes enable this approach. Fortunately, there are many alternatives for scaling these activities to share the load and achieve economies of scale for all digital preservation practitioners to benefit from.

This use of derivative environments, instead of capturing full copies of the entire environment each time a change is made, has two major benefits that may enable scaling of the use of EaaS in the future. Firstly, it minimizes the additional data storage required to maintain large collections of custom content environments (environments pre-configured to load content such as a single file or set of files on execution). Secondly, it further simplifies and extends the digital preservation community's options for scaling and providing software and content services using emulation and lowers the storage requirements while doing so. It also offers potentially advantageous opportunities for the seamless provision of services software and emulation tools across multiple organizations. For example, this approach could enable one provider to host base environments, another to host the software-layer derivatives that point to the base environments, and a third to host the content environment derivatives that point to the software environments. All of those layers could then be brought together using an EaaS implementation hosted by an additional provider and all of this could happen seamlessly and in real time from the end-user's perspective.

The UNESCO PERSIST project [6], The Software Heritage Foundation [7], and The Software Preservation Network [8] are working to enable access to software archives. Using the EaaS functionality that enables the layering of derivatives, and their local preservation environments, they could potentially each play a role in providing software for use in emulation solutions without requiring end-users to each preserve the software themselves.

The involvement of these larger bodies promises the possibility of a solution to one of the largest outstanding issues, the permission required by the Intellectual Property owner of the original operating system and software to operate the system. [3] By encouraging these IP owners to allow access to older versions of their software under given licence terms, EaaS becomes a commercial reality. The licence terms are likely to dictate who can use the software and under which circumstances, and a centralised service could enforce these licence terms in a consistent manner to the satisfaction of the IP owner.

Applying the semi-curated approach YUL has implemented for it's 6,000-10,000 CD-ROMs would not be sustainable for a larger scale use of EaaS for providing access to single files. In contrast migration enables content to be reused by ensuring it is available in file formats that modern software can interact with, and Preservica already provides comprehensive tools to automate migration at scale. However, emulation is sometimes necessary e.g. where custom software is involved, such as is the case with many of YULs CDs. In other cases, emulation is desirable due to its ability to provide access to the content using original software,

[6] UNSECO. 2016. PERSIST: UNESCO Digital Strategy for Information Sustainability, (2016). Retrieved March 31st, 2017 from http://bit.do/UNSECODigitalStrategy

[7] Software Heritage. 2017. "Software Heritage Foundation", (2017). Retrieved March 31st, 2017 from https://www.softwareheritage.org/

[8] Software Preservation Network, 2015. "About [the Software Preservation Network]", 2017. Retrieved March 31st, 2017 from http://www.softwarepreservationnetwork.org/about/

ensuring rendering or interaction issues are minimized [9]. Unfortunately, manually creating custom content-environments for each file in a collection of millions of files would not be viable due to the cost. However, a future potential large-scale application of EaaS would be to implement it as an automated "universal viewer" within preservation and access systems. This approach might entail file format information being provided to EaaS from a digital preservation system and EaaS automatically associating the file with an appropriate preconfigured software environment. To enable this the digital preservation system would have to be able to provide the file format information via its API. Alternatively, the digital preservation system could do the matching with an external software-environment library during the process of ingesting the file into the digital preservation system. In that scenario, the digital preservation system could then just provide the content file and the identifier for the environment needed to interact with it via its API to EaaS. The environment could then be edited in real time to wrap the file in a disk image, attach the image to the environment and insert a link into e.g. the Windows start-up folder (in the case of a Microsoft Windows-based software environment) on the main emulated hard drive. This would force the system to automatically open the file when the emulated computer was loaded. Scaled across many software applications this approach thus has the potential to provide a 'universal viewer' service.

Within Preservica, the 'Universal Viewer' application of emulation technology could also be integrated within the system by including the (open source) EaaS framework as a Preservica toolset to sit alongside other simple file viewers. In Preservica processes can be scaled horizontally (enabling multiple processes to run concurrently) by adding additional processing servers. This is currently a simple but semi-manual step. If EaaS were integrated more tightly into Preservica a dynamic scaling mechanism may be required such as the ability to spin up additional processing servers in the cloud on demand for use in emulating additional computers [10]. However, investment to achieve that result may be advantageous for a number of reasons. In particular the association between software environments and file formats, and the preservation of derivative environments and other assets in the emulation stack could be more tightly integrated and seamlessly automated within the Preservica system. The addition of technology seamlessly linking EaaS technology into Digital Preservation products such as Preservica is dependent on a resolution of the licencing issue of the operating systems and software packages being run.

5 CONCLUSION

The marrying of EaaS and Preservica at Yale University has enabled Yale University Library to ensure its content can be made accessible via emulation while simultaneously being preserved in a robust digital preservation system. In addition, the base images, software and derivatives can all also be preserved in a trustworthy digital preservation system with little additional staff effort. The work to reach this point has been significant but the existing tools, Preservica and bwFLA Emulation as a Service, with their generic APIs and sophisticated out-of-the-box functionality, have made much of this integration relatively straightforward to implement.

There are still gaps in the emulation and software preservation services landscape that will prevent less well-resourced organizations from implementing a similar approach. Fortunately, the features available in EaaS imply clear pathways to implementing software archives as a service, pathways that may be palatable to the software IP owners. In addition, the benefits of pursuing this and applying EaaS to create a "Universal Viewer" are hopefully obvious to all.

Further integration of emulation services into digital preservation systems such as Preservica seems inevitable. Fortunately, their existing functionality, such as Preservica's preservation metadata registry, standard APIs, and ability to scale horizontally, ought to make this a reasonably straight forward process.

REFERENCES

[1] David Bearman. 1999. Reality and Chimeras in the Preservation of Electronic Records. *D-Lib Magazine* 5, 4 (1999).

[2] Klaus Rechert, Isgandar Valizada, Dirk von Suchodoletz, and Johann Latocha. 2012. bwFLA – A Functional Approach to Digital Preservation. *PIK – Praxis der Informationsverarbeitung und Kommunikation* 35, 4 (2012), 259–267.

[3] Richard S Whitt. 2017. " Through A Glass, Darkly" Technical, Policy, and Financial Actions to Avert the Coming Digital Dark Ages. *Santa Clara High Technology Law Journal* 33, 2 (2017), 117.

[9] See: Cochrane, E. 2012. "Rendering Matters - Report on the results of research into digital object rendering". Archives New Zealand January 3, 2012. Retreived March 31st from: http://archives.govt.nz/rendering-matters-report-results-research-digital-objectrendering

[10] Note that emulator hardware demands are fairly light when emulating older computers. Multiple emulated computers can share one CPU thread and memory (RAM) requirements are insignificant (e.g. 16 Megabytes for 1990s era IBM-compatible PCs)