A FRAMEWORK FOR AGGREGATING PRIVATE AND PUBLIC WEB ARCHIVES

by

Matthew R. Kelly
B.S. June 2006, University of Florida
M.S. May 2012, Old Dominion University

A Candidacy Proposal Submitted to
the Faculty of Old Dominion University

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
July 2018

Approved by:

Michele C. Weigle (Director)

Michael L. Nelson (Member)

Danella Zhao (Member)

ABSTRACT

A FRAMEWORK FOR AGGREGATING PRIVATE AND PUBLIC WEB ARCHIVES

Matthew R. Kelly
Old Dominion University, 2018
Director: Dr. Michele C. Weigle

Web archives preserve the live Web for posterity, but the content on the Web one cares about may not be preserved. The ability to access this content in the future requires the assurance that those sites will continue to exist on the Web until the content is requested and that the content will remain accessible. It is ultimately the responsibility of the individual to preserve this content but attempting to replay personally preserved pages segregates captures by individuals and organizations of personal, private, and public Web content. This is misrepresentative of the Web as it was. While Memento may be used for inter-archive aggregation, no dynamics exist for the special consideration needed for the contents of these personal and private captures.

In this work we introduce a framework for aggregating private and public Web archives. We introduce three "mementities" that serve the roles of the aforementioned aggregation, access control to personal Web archives, and negotiation of Web archives in dimensions beyond time, inclusive of the dimension of privacy. These three mementities serve as the foundation of the Mementity Framework. We investigate the difficulties and dynamics of preserving, replaying, aggregating, propagating, and collaborating with live Web captures of personal and private content. We offer a systematic solution to these outstanding issues through the application of the Framework. We propose a research plan for the outstanding issues to be explored in this proposal to ensure the Framework's applicability beyond the use cases we describe as well as the extensibility of reusing the mementities for currently unforeseen access patterns. We outline an evaluation plan to formally justify the mementity design decisions, quantify the anticipated temporal and spatial costs, and evaluate the feasibility of the dynamics described through software-based implementations.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

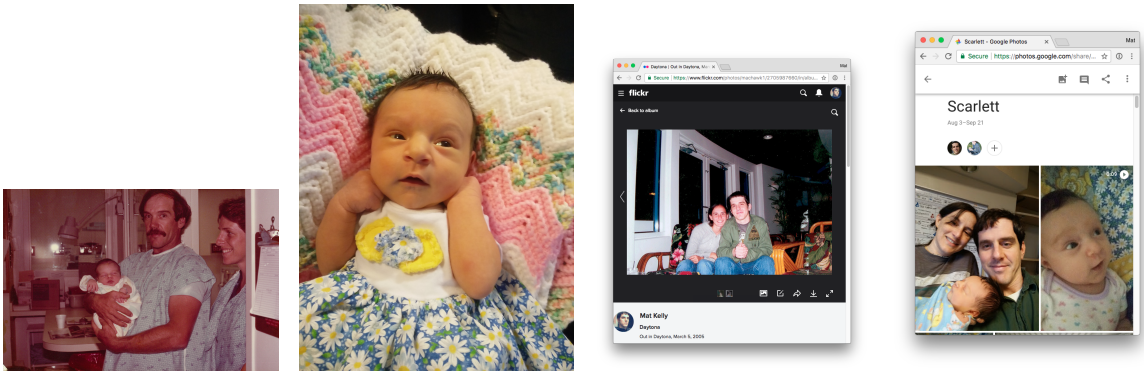
CHAPTER 1

INTRODUCTION

> *The past few decades have witnessed the demise of numerous forms of digital storage. This has prompted my observation that digital information lasts forever — or five years, whichever comes first.*
>
> - Jeff Rothenberg, *Ensuring the Longevity of Digital Information [151]*

Society looks to the Web as a source of up-to-date information, a repository for personal expression, and a record of the past. Unlike analog records like a newspaper or a physical photo book, the Web is an ephemeral medium. The ephemerality of content on the Web becomes particularly important to a Web user when the content serves as a personal record. For example, surfacing baby photos posted long ago from Web sites like Facebook[1] or Google Photos[2] (scans of my own shown in Figure 1) requires a degree of accessibility not currently present in these services. In the case of these sites, precise dates or non-linear traversal (e.g., Facebook uses a temporal range on-demand model as in Figure 2) are required to efficiently locate a photo or a post in time among the potentially plethora of other posts that have accumulated. The ability to access the photos on these sites in the future also requires the assurance that those sites will continue to exist on the Web until the content is requested and that the content will remain accessible [116].

[1] `https://facebook.com`
[2] `https://photos.google.com`



(a) **(b)** **(c)** **(d)**

Fig. 1: (a) My baby photo (digitally scanned) has persisted because my parents and I have been been the bearers and responsible for its continued persistence and accessibility. As the responsibility for the photo's persistence moves from my retaining a physical copy to photos residing solely on a Web site, I can no longer be certain the content will be accessible in the future. (b) A born-digital photo of my daughter without an analog version. (c) A photo I digitally scanned and uploaded to Flickr in 2005 only to view it again on the live Web in 2017. (d) gives context to (b) to be within a digital album on Google Photos (now the bearer).

A paradigm for recalling personal photos before the Web required a responsible party to physically retain them. Figure 1a shows an example where I, as the *bearer* of the physical or "analog" version of the photo, was able to surface and scan the image for uploading to the Web. The preservation and continued accessibility of the content when it was a physical object was dependent upon me. Reassigning the bearer role at the time of digitization of the analog content to another entity (e.g., Facebook or Google Photos) puts the task of ensuring posterity of the content into the hands of an entity that is not me. This new bearer may not deem the content as important as the hard-copy baby photos are to me. Were these photos posted to Facebook and naïvely assumed as safeguarded [116, 110], I or someone interested in the photos remaining accessible for posterity (e.g., my parents) may wish to take a further action to facilitate them remaining accessible. This becomes particularly important when the original photo is born-digital, e.g., the photo of my daughter in Figure 1b solely resides within an album (Figure 1d) of an external bearer (Google Photos).

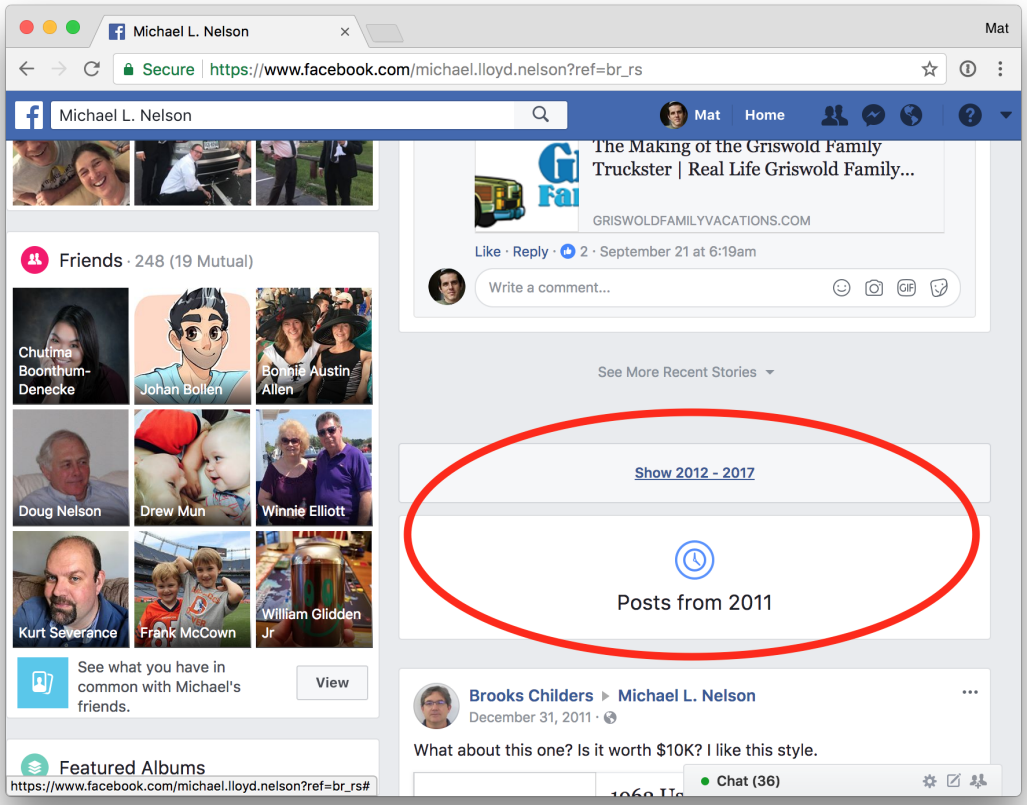


Fig. 2: Photos and posts on Facebook are not necessarily linearly displayed in temporal order, requiring a drill-down approach with recollection of when an event was posted to surface content. Circled is the option to drill down further. Selecting this option also obscures other temporal ranges, only providing data for the part of the range selected.

The World Wide Web is more ephemeral than analog mediums like books, newspapers, journals, and hard-copy baby photos [122]. Web media may change its contents, layout [46, 47], and accessibility [48, 105] on subsequent viewings [88]. Archival Web crawlers like Internet Archive's Heritrix [127] can be used to capture the content on the Web at a specific point in time. Many Web archives (e.g., the British Library's UK Web Archive[3]) use Heritrix to preserve the part of the Web relevant to their interest (e.g., content within a specific TLD [59] or about a specific topic) for posterity. Individual archivists may also use this software or other existing Web

[3] `https://www.webarchive.org.uk/ukwa/`

TABLE I: Both personal and institutional Web archiving can be either public or private. Listed here are scenarios where this would occur.

	personal	**institutional**
private	My Facebook.com feed	Corporate Intranet [45]
public	What I see at `cnn.com`	Large-scale Web crawls (e.g., IA)

archiving software to preserve content to a personally owned file store. With individuals' ability to preserve their own content on the Web, the act of doing so may seem like a solved problem. However, other issues remain for the interoperability, privacy, and accessibility of the preserved content that requires further investigation as more personal and private content on the live Web proliferates.

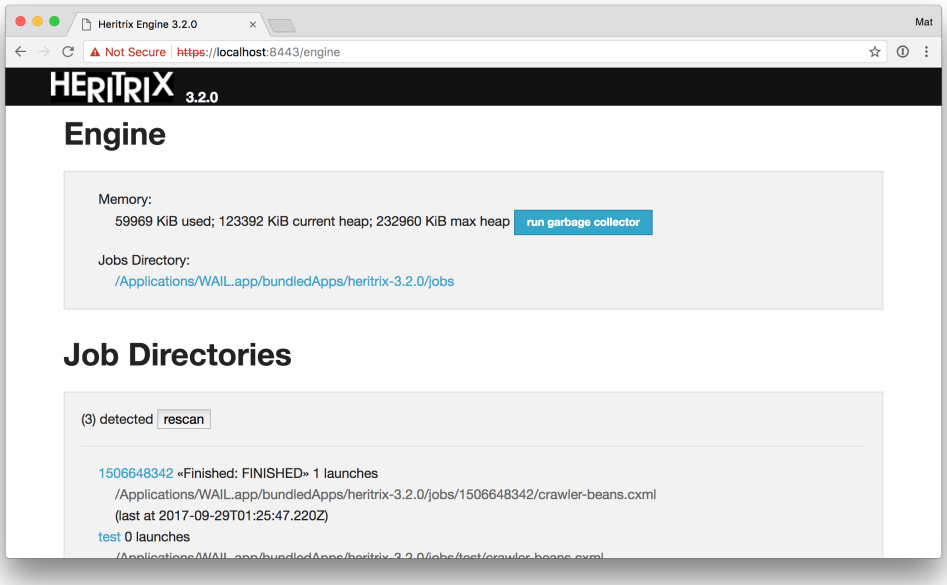
In this work, *personal Web archiving* constitutes Web archiving by an individual without restriction to the availability of content (e.g., private or public) on the live Web. For example, Alice archives what *she* sees on the live Web of a publicly accessible `cnn.com`, her private `facebook.com` news feed, and her publicly accessible but not well-archived vacation photos Web site. This can be compared to *institutional Web archiving*, which is performed by an organization with the goal of long-term preservation. Web archiving by an institution need not be of public content; for example, an institution may perform large-scale preservation of their Web presence partially consisting of content behind authentication. While most institutional efforts focus on the publicly available live Web, institutional Web archiving, much like personal Web archiving, is not limited to the availability of the content (e.g., contains access restrictions) on the live Web. In contrast to institutional and personal Web archiving, *private* and *public Web archiving* define the availability of the content to be preserved as it exists on the live Web. Table I shows example of each of the four permutations of personal/institutional and public/private Web archiving. Personal Web archives may contain representations and resources from either or both of the publicly available or private (e.g., behind authentication) Web. Additionally, both personal and private Web archives may contain personalized content like cookies and session information [23] obtained at crawl time as well as GeoIP-dependent rendering of Web pages [99]. Institutional Web archiving mostly focuses on the publicly available live Web but is not inherently limited to this [22, 72]. Table II describes the bounds of some features of Web archives like who administrates

TABLE II: A variety of features can be used to classify Web archiving efforts.

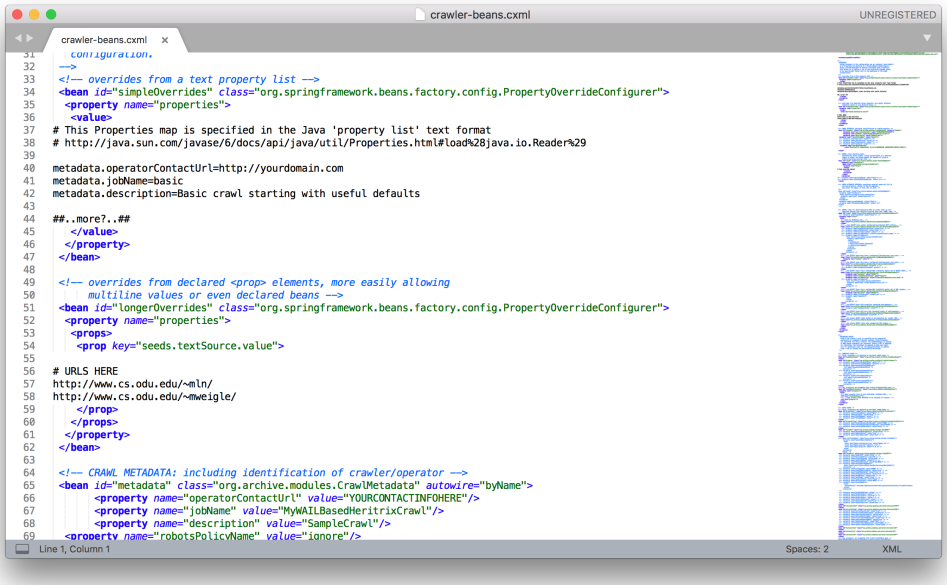
individual	←	administration	→	organization
personal	←	scale	→	institutional
targeted	←	capture scope	→	open
personalized	←	capture session	→	public
restricted Web	←	crawler time access	→	public Web
restricted	←	replay perspective	→	public

the archive (individual vs. organization), the size or scale of the archive (personal or institutional), the scope of the capture procedure (open vs. targeted), the nature of the preserved content (personalized vs. public), the accessibility of the content at crawl time (restricted or publicly available), and the accessibility when the capture is re-experienced (restricted vs. public).

Because of the technical requirements and know-how required to use Heritrix (Figure 3), few users archive their content from the live Web using standard good practice but instead resort to easier, often ad hoc methods [173]. The standardized formats created for Web archiving (e.g., WARC [82]) provide most of the structure needed to portably store content for longevity. Only recently has the means for a non-technical user to produce personal Web archives [28, 107, 143] in this format begun to come to fruition. Despite this, were users able to preserve their Web data in a standard form, most would still be unaware of how to access and control access to their personally-archived sensitive and non-sensitive information.



(a)



(b)

Fig. 3: While the Heritrix user interface is intuitive to manage the state of existing or predefined crawls (a), extensive training is required to get to this point. Furthermore, without an interface to configure new crawls, users may need to manipulate an XML template (b) to obtain the results they desire from the crawl.

Collaboration and access control are rarely considered in contemporary Web archiving due to the majority of Web archiving efforts targeting publicly-accessible Web content. A reason for this is that the act of preserving Web content by individuals is met with scrutiny of authenticity, i.e., content may have been manipulated prior to capture. Captures performed by those without vetting are not afforded to the degree of authenticity as institutional captures [84]. However, it remains that individuals preserving content they deem important (like one's baby photos on Google Photos), regardless of vetting, ought to be preserved so as to not be lost in time – even if the representation has potentially been manipulated. Content that is preserved by individuals needs further consideration for access control, as well as authenticity, if it is to publicly stand as a capture in the historical record.

The remainder of this chapter provides examples where the framework we propose would facilitate personal and private Web archiving in a systematic way that considers privacy and access control. Section 1.1 describes current methods and needs for personal Web archiving by various organizations and individuals. Section 1.2 highlights the preservation of content missed by archival efforts such as Web content that requires authentication. Section 1.3 discusses preservation of content that requires access control at preservation time, replay time, and when collaborating or disseminating Web archives. Section 1.4 outlines the organization of the following chapters in this research proposal.

1.1 IT WAS THERE YESTERDAY, WHERE DID IT GO?

During the Boston Marathon and London subway bombings of 2013 and 2005 (respectively), digital humanities researchers sought to capture relevant social media Web pages at Reddit[4], Imgur[5], and Twitter[6] [125, 138, 137]. In data collection procedures previously performed by the researchers, the group captured this content through screenshots — a manual and labor-intensive process that did not yield captures with the flexibility of other Web archiving formats. These screen captures are not interactive like the original and provide no context of linkage to other relevant documents preserved at the same time. Traditional Web archiving tools like Heritrix are not equipped to quickly respond to rapidly changing conditions to capture Web

[4] http://reddit.com
[5] http://imgur.com
[6] http://twitter.com

pages as conversations are occurring [173].

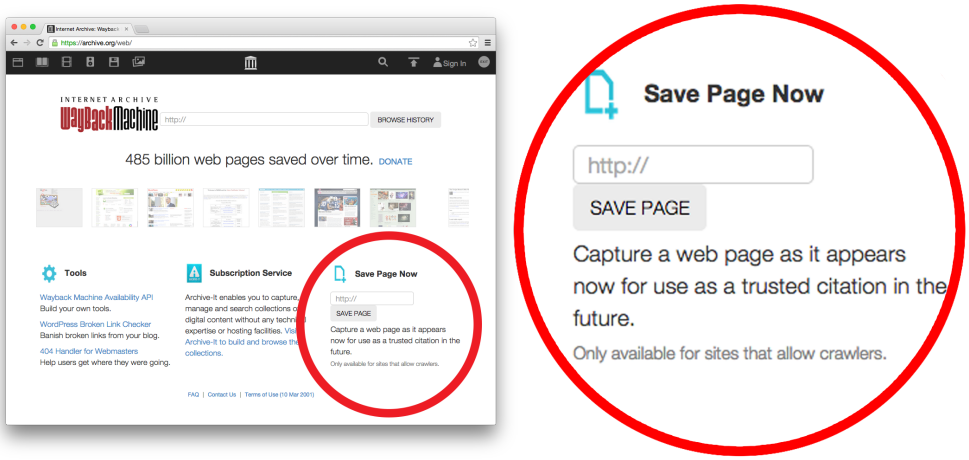


Fig. 4: In 2013 [150], the Internet Archive (pictured on the left) began allowing users to submit URIs for Web sites (through the interface pictured on the right) to be preserved. The resulting Web archives are retained on their server and are accessible to the user. This approach also exhibits the URI collision problem (Section 1.2), the inability to preserve content that requires authentication, and a slew of other personal Web archiving issues that are inherent in using institutional archives.

Many Web users naïvely assume that the content they view on the live Web is in little danger of disappearing [116]. The Internet Archive[7], an institution set up to preserve content on the public Web, has frequently served as a safeguard for those that believed this assumption [123]. Threats toward the longevity of archives include both technical failures (e.g., software, hardware, media) as well as non-technical (e.g., natural disasters, economic failure) [149]. In large, the Internet Archive has a "collect everything" best-effort collection development policy. In 2013, the Internet Archive began providing a Web-based submission form for users to submit a capture of a single URI[8] (Figure 4) [150]. Relatively obscure and personally important content is less likely to be saved for future viewing than popular Web pages [4]. The proactive approach for a user to "preserve" a Web page is to simply take a screenshot of the Web page. This approach causes the collection of captures to quickly become outdated as new content is added [116] and is difficult to query and access without a large amount of curation. This level of curation of providing metadata (e.g., the original URI, datetime of capture, or to which collection an archival crawl or capture

[7] `http://archive.org`

[8] This also includes any embedded resources on a Web page such as images, JavaScript files, etc.

belongs) for accessing captures of Web content exists in the software implementation of Heritrix and the WARC format [82]. However, this format is limited in accessibility for interaction by end-users and is meant more to be produced and consumed by software rather than interacted with directly like saved HTML or a screenshot of the Web page.

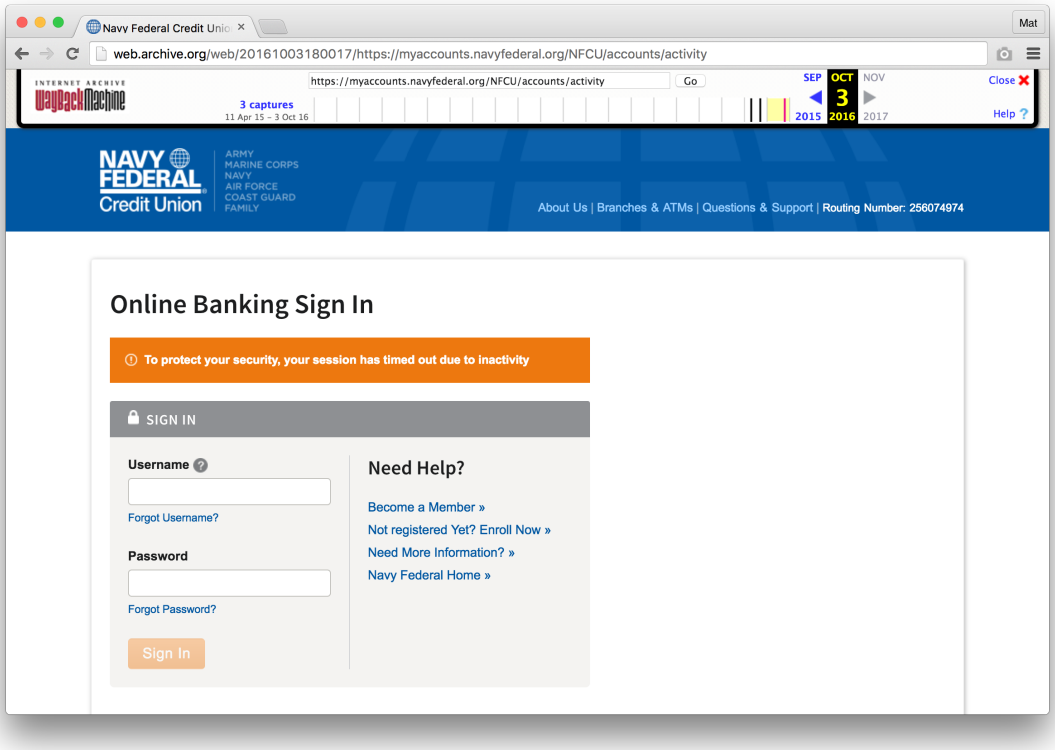
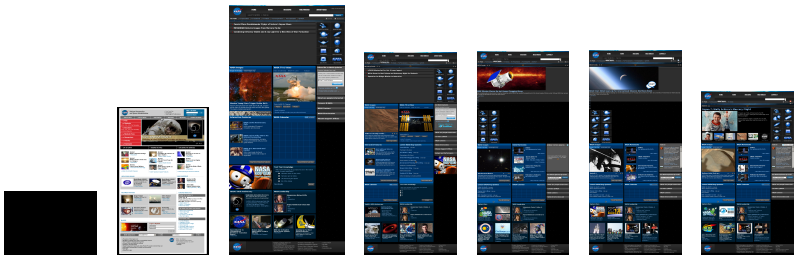


Fig. 5: A user attempting to naïvely preserve their account information or any content behind authentication frequently receives a preserved login screen. Submitting a URI to be preserved from either an institutional archive's Web interface or even to an archival crawler on the user's own machine for local preservation is insufficient context to preserve content behind authentication.

Comprehensively collecting all data required to replicate the full experience of replaying the live Web site once archived is tedious and error-prone and thus the process is usually tasked to a programmatic script or crawler. State-of-the-art archival crawlers are limited in what they can capture behind authentication on the Web, so even using institutional grade archiving tools would likely not adequately archive the Web content (Figure 5) [18]. As an additional caveat, the difference in access mechanism (archival crawler instead of a user's browser) makes it unlikely that the

(a) 1996 **(b)** 1997 **(c)** 1998 **(d)** 1999 **(e)** 2000 **(f)** 2001 **(g)** 2002 **(h)** 2003 **(i)** 2004 **(j)** 2005

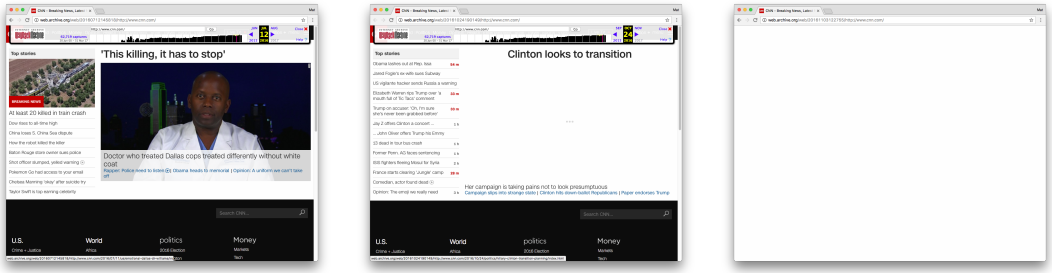


(k) 2006 **(l)** 2007 **(m)** 2008 **(n)** 2009 **(o)** 2010 **(p)** 2011 **(q)** 2012

Fig. 6: NASA over time. Changes in design and thus the technologies used is easily observable between 1997 and 1998, 2002 and 2003, 2006 and 2007, and 2007 and 2008. The captures from 2003 to 2006 appear completely black due to the difference in the archival crawler's capability compared to the technology that resided on the page in this time range [100].

same content that the user wishes to preserve can and will be captured. Archival crawlers often lag behind Web standards and thus Web pages that implement those standards. Because of this, attempts to preserve a page using technology beyond the capability of an archival crawler but perfectly inline with contemporary browsers' capabilities causes content that appeared as expected in a browser to not be captured by the crawler. An example of the functional difference between archival crawlers and Web browsers over time can be observed in annual captures of `nasa.gov` [100] (Figure 6), which went through a phase (2003-2006) where content was viewable on the live Web but unable to be archived by the crawlers at the time. The problem is not only one of the past, but is recurring. A recent example is of `cnn.com` being preserved by Internet Archive [26]. When a user re-experiences the page through the archive's replay system, the system executes the archived representation of the live Web `cnn.com`'s JavaScript (Figure 7c). This JavaScript programmatically assumes it is on the live Web and prevents the page from being displayed.

This potential for uncertainty in the reliability of the capture is not limited to content behind authentication. We can see how `cnn.com` looked in September 2016 (Figure 7a), but some captures are incomplete (Figure 7b) or contain errors that prevent the page from rendering at all (Figure 7c) [26, 27]. Given the lack of confidence in knowing the completeness of captures without comprehensively dereferencing all captures' identifiers (URIs), users may question the accuracy of the historical record.



(a) **(b)** **(c)**

Fig. 7: The quality of the capture over time for three different archived representations for `cnn.com` shows (a) a visually complete[1], (b) visually damaged[2], and (c) very damaged[3] representation.

1.2 SAVE THIS, BUT ONLY FOR ME

A large part of the content on the Web requires authentication for access and thus is largely inaccessible to Web archiving software – sometimes for good reason (e.g., unsuitability of preservation) and other times by technical limitations of the software [105, 100]. The dynamics of Web applications compared to static Web pages introduces an additional degree of dimensionality into the problem with URIs "colliding". One scenario where URI collision occurs is when content behind authentication is co-located at the same URI as publicly available content. For example, Figure 10 shows `facebook.com` as captured by an individual (preserving content behind authentication) and the same URI captured by Internet Archive (only the login page was preserved). While additional parameters (e.g., cookies, session identifiers [23]) provide a way to distinguish content on the live Web, these supplemental access entities do not carry over nor are they suitable when viewing preserved content at a

[1] `http://web.archive.org/web/20160712145818/http://www.cnn.com`
[2] `http://web.archive.org/web/20161024190149/http://www.cnn.com`
[3] `http://web.archive.org/web/20161103122755/http://www.cnn.com`

later date in the archives (e.g., cookies would be long expired upon access). With the intermingling of captures of what I, other individuals, and institutions each saw, it would be important to give precedence on the representation of the perspective of the Web as viewed. There is no single correct representation (e.g., what both I and a crawler saw simultaneously existed at the same URI), but there are also no semantics to express preference of the representation beyond URI and datetime.

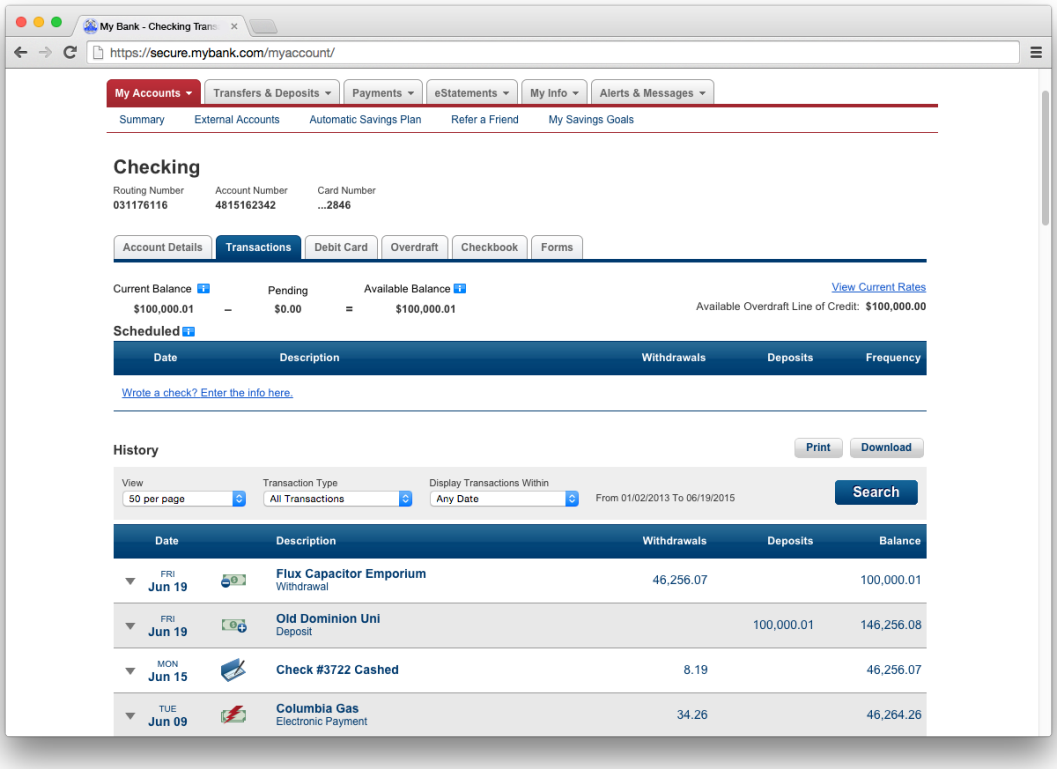
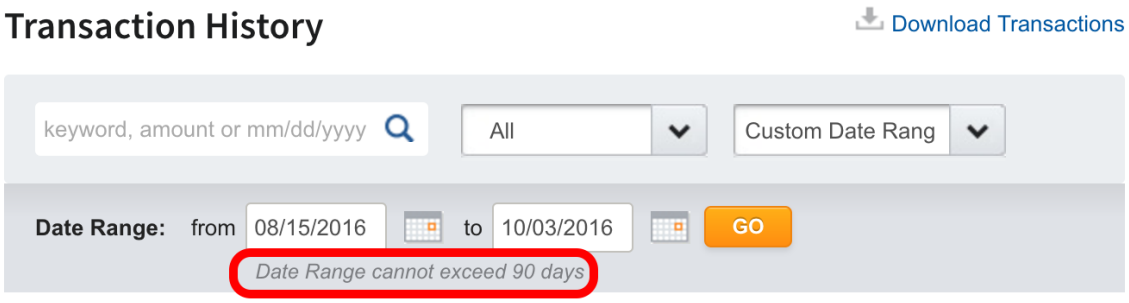


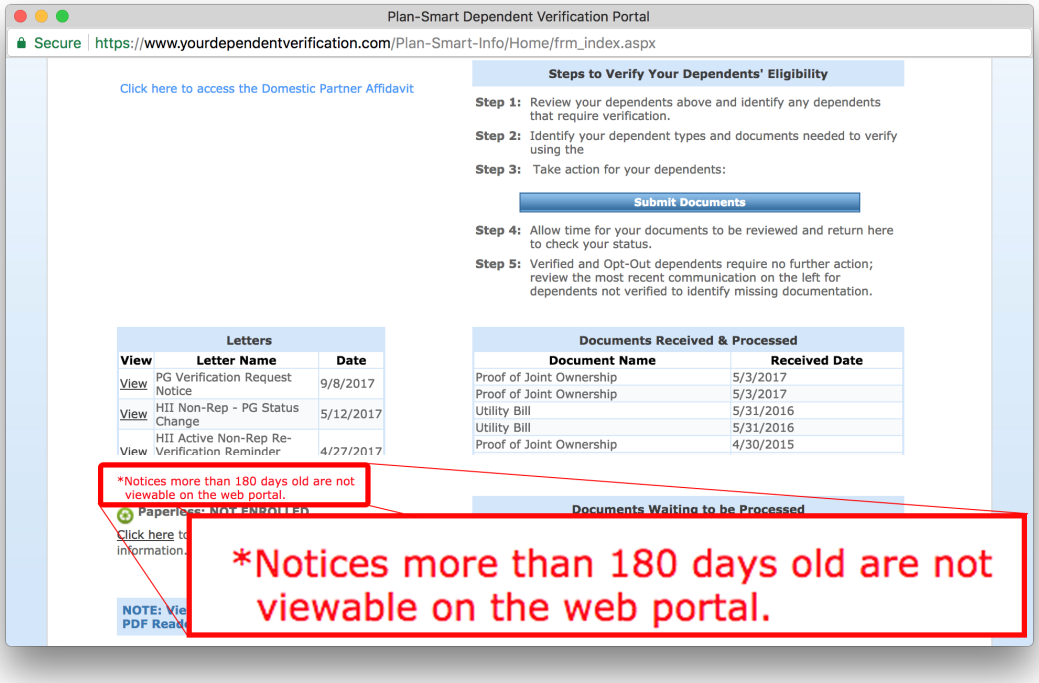
Fig. 8: An online bank account statement is an example of private content on the Web that one might wish to preserve but not publicly share.

As another example, banks frequently encourage clients to "go paperless", allowing their bank records to be accessed on the bank's Web site (Figure 8) while foregoing paper statements. Oftentimes, banks' sites limit how far in a user's bank history the user may access and how much of the history can be accessed at once (Figure 9a). A client wanting to preserve this history in its original form to recall history beyond the limit of what the bank's live Web site currently allows may save the Web page or take a screenshot of the page. However, in doing this, any interaction within the page becomes unusable, as secondary data (e.g., images of paper checks)

may not be exposed with these ad hoc methods. This restriction pattern also exists in other organizations that provide soft copies of documents. Figure 9b shows an online verification service that not only limits access to a span at a time, like Figure 9a, but also completely removes access after 180 days. This sort of ephemerality mimics the conventional loss of access of conventional resources representation on the public live Web.



(a) Online banking restricts how much account history a user may obtain at a time.



(b) Another organization with a temporal limitation of access.

Fig. 9: Banks frequently limit how far back in the history of an account (circled in red in (a)) and the quantity of data available for viewing at a time, exacerbating personal offline preservation of this data by the individuals who own the account. This behavior is not limited to banks, however, as other organizations that provide digitized statements (b) also remove access to the account holder in time.

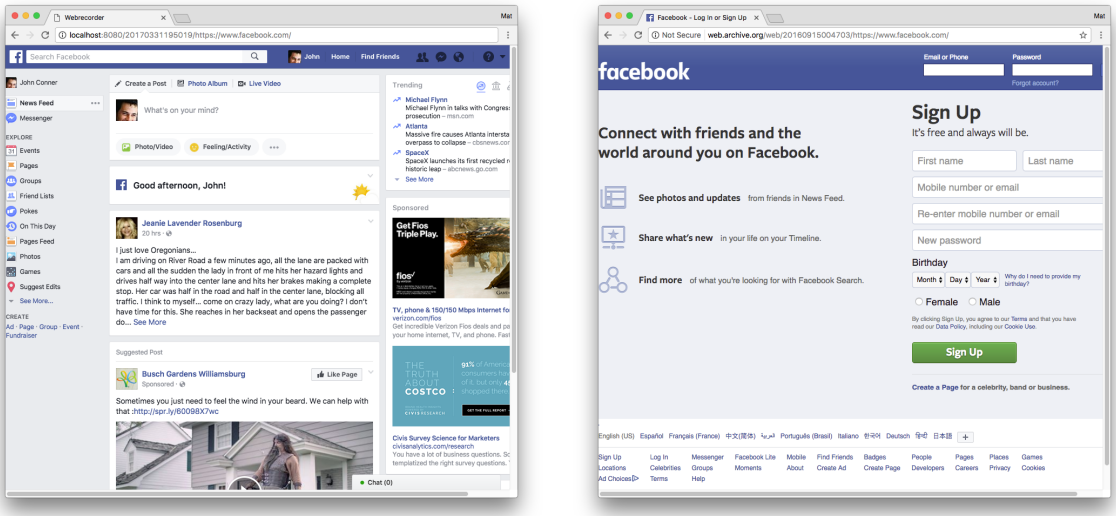
1.3 I WANT TO SHARE THIS BUT CONTROL WHO CAN SEE IT

The MITRE Corporation is a not-for-profit corporation that operates multiple Federally Funded Research and Development Centers (FFRDCs) on behalf of the

US Federal Government to address the nation's toughest challenges [126]. MITRE sought to automatically archive their corporate intranet using Web-scale Web archiving tools [45]. Certain sensitive content on the intranet required credentials to be accessed, some of which was enforced via JavaScript, which made archiving the content unreliable due to the functional shortfalls of archival crawlers. MITRE's requirement to responsibly manage data including misplaced and misclassified data required a "clean-up" procedure of the archive prior to making the archive accessible within the corporation. This procedure incurred collateral damage, causing content stored in the same WARC as sensitive information to also be wiped. A more sophisticated approach would be to preserve content for access by only those with the appropriate access to view the data as it resided on the live Web.

Another scenario where access control is needed on the archived Web is in ensuring that the access control that universities and organizations with privileged or paid access to resources, e.g., an online academic journal subscription, is maintained when the content is archived and replayed. Having a framework in-place to facilitate this would encourage reuse and establish integrity of the data as well as increase the availability of the data were the original source on the live Web moved or deleted.

As personal and private Web archives proliferate and users proactively preserve their content from the live Web, their personal Web archives may contain captures with personally identifiable or sensitive information (e.g., their `facebook.com` feed, Figure 10a). A user may want to selectively share their captures but wish to also regulate access to their captures. Without the context of authenticating as a user, many archives simply preserve the login page (Figure 10b). Both captures are representative of `facebook.com`, potentially even captured at the same time. Without context for the capture of `facebook.com` to reliably re-experience what they preserved and a mechanism to regulate the capture in Figure 10a, users may be hesitant to share and propagate their captures [120].



(a) Local Archive capture **(b)** Internet Archive capture

Fig. 10: `facebook.com` as captured by an individual versus an institutional Web archive.

1.4 RESEARCH QUESTIONS

In this proposal we will investigate a framework to mitigate the outstanding issues relating to private, public, and personal Web archiving. The goal of this research is to provide strategic practices, technologies, and hierarchies for systematically replicating the live Web, particularly inclusive of the parts that currently are not preserved.

Based on the issues previously described, we wish to address the following research questions:

RQ1: What sort of content is difficult to capture and replay for preservation from the perspective of a Web browser?

RQ2: How do Web browser APIs compare in potential functionality to the capabilities of archival crawlers?

RQ3: What issues exist for capturing and replaying content behind authentication?

RQ4: How can content that was captured behind authentication signal to Web archive replay systems that it requires special handling?

RQ5: How can Memento aggregators indicate that private Web archive content requires special handling to be replayed, despite being aggregated with publicly available Web archive content?

RQ6: What kinds of access control do users who create private Web archives need to regulate access to their archives?

Chapter 2 describes the foundational knowledge of the Web and Web archiving. Chapter 3 describes the state of the art in Web archiving practice and current efforts and research investigating archiving topics like migration, security, collaboration, etc. of the archived data. In Chapter 3 we define the nomenclature that serves as the foundation for this research to be described in later chapters. Chapter 4 describes preliminary research we have performed to address problems in Web archiving that we have encountered in the initial stages of this research. In Chapter 5, we describe the entities that play a role in the framework of allowing private and personal Web archives to be systematically replayed with access control in a way that encourages replication of the original live Web experience. In this chapter we will also detail how we will evaluate the framework both qualitatively and quantitatively. In Chapter 6, we propose a work schedule for when this research described will be completed and expound on how this work will address the research questions enumerated above.

CHAPTER 2

BACKGROUND INFORMATION

> *...we see more and farther than our predecessors, not because we have keener vision or greater height, but because we are lifted up and borne aloft on their gigantic stature.*
>
> - John of Salisbury, *The Metalogicon*

In this chapter we describe prior relevant work and concepts relating to the Web and Web archiving.

2.1 THE WEB

Tim Berners-Lee described what we know as the Web [32] as a system of clients communicating with servers. In addition to accessing other resources on the server itself, servers could also reference resources on other servers through addressing. The Hypertext Transfer Protocol (HTTP) was initially created by Berners-Lee et al. [29] and refined by Fielding et al. [61]. The latter (HTTP 1.1) accounted for some of the initial protocol's shortcomings like the inability to establish persistent connections, the lack of explicit requirement of a Host header in a request, etc. In 2014, Fielding et al. partitioned the specification into six separate RFCs [66, 67, 65, 62, 63, 64] to more cohesively describe each feature of the protocol with more clarity and less repetition in separate documents. The protocol has since been optimized for more efficient pipelining of communication and secure transfer with HTTP/2 [24]. However, Berners-Lee's seminal description of the Web as a relationship between resources and their representations is critical to understand as a foundational concept in our work.

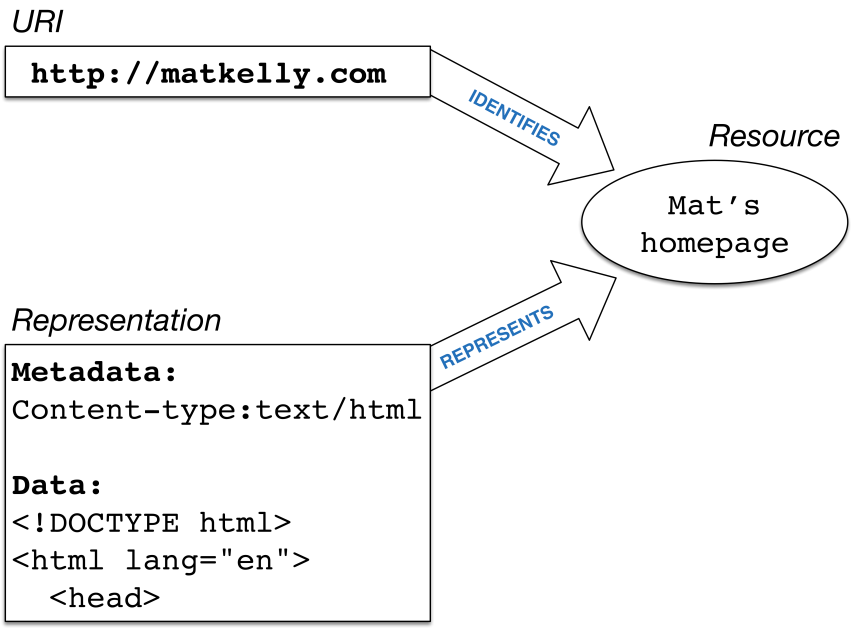


Fig. 11: Sample relation between a URI, resource, and representation on the Web.

Communication using HTTP entails a series of HTTP requests and HTTP responses. Uniform Resource Identifiers (URIs) identify Web resources without being bound to the resource's type or current accessibility [31, 33, 114, 36]. When a URI on the Web is dereferenced, a representation of the resource is returned (Figure 11). Upon a client dereferencing a URI from a Web server, the server responds with HTTP headers preceding and describing the content to be subsequently delivered (Figure 12). These headers consist of an HTTP response status code [67] indicative of the server's success on being able to deliver a representation for the resource, the willingness of a server to respond with the content requested, etc. Other metadata about the response like the Content-Type, Date, and Server information is also provided in these headers [37].

```
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 02 May 2017 16:13:33 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="style.css" media="all" />
...
```

Fig. 12: Sample HTTP Response from the URI-R `http://matkelly.com`.

```
HTTP/1.1 200 OK
Server: Apache
Date: Wed, 03 May 2017 12:01:10 GMT
Content-Type: text/html
Link: <http://mybook.com/toc>; rel="contents",
↪ <http://mybook.com/pages/246.html>; rel="next last",
↪ <http://mybook.com/acks.html>; rel="section
↪ http://mybook.com/myrelations/acknowledgements"
```

Fig. 13: The values for the Link HTTP Response header may be derived from a registry of valid values [81] or a URI [130]. Each link is comma delimited and each link-value is space delimited. The third link shown specifies an extension relation using a URI.

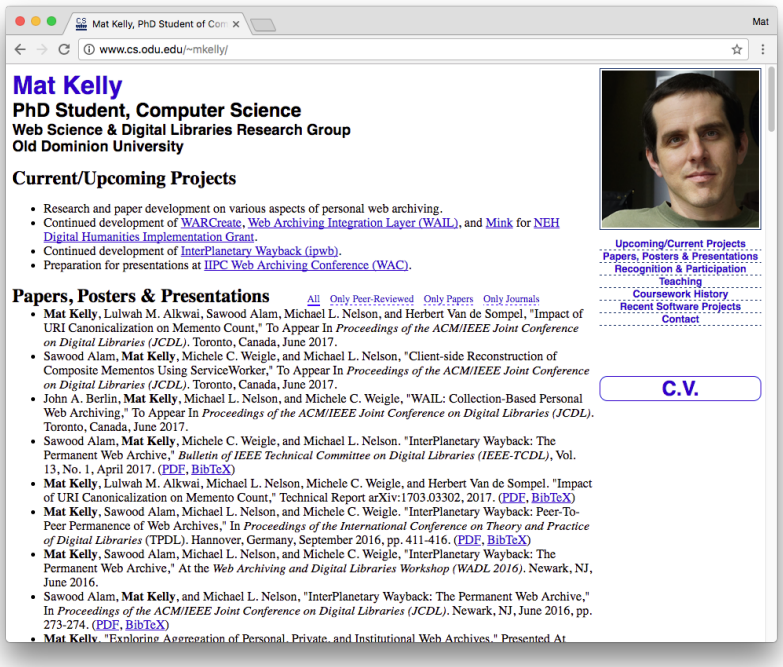
Attributing the relationships between resources on the Web allows expression of resource association. Nottingham (in initial [130] and more recently updated [132] specifications) defined how to represent relations between URIs through use of the `Link` HTTP header. Using a standard yet extensible syntax, resource representations may specify other URIs that relate to either the resource itself or give context

of the URI relative to other identifiers listed. This context is described using a value (`link-value`) for the relation type, defined within the "rel" attribute, associated with a URI. While the Web Linking specification establishes a registry [81] containing the recognized relation types and their semantics, it also allows for "extension relation types". Extension relations are defined with a URI [31] as the value for the respective `rel` attribute. For example, Figure 13 shows an HTTP response for a request for an online book with a Link response header containing three links. While the first two links are straightforward, specifying the table of contents and the coinciding next and last page of the book, the third relation specifies an extension relation type, presumably of the book's acknowledgements section (though to infer or assume semantics from the URI is fallacious [35, 131, 144]). Preservation of the live Web requires maintaining the relation between the archived representations and the original representation on the live Web. The Memento Framework builds heavily on Nottingham's Web Linking specification and is discussed for relevance to this research in Section 2.5.

URIs that identify Web resources should remain stable, or "cool", and should not contain the mechanism of how a server is run (e.g., a `cgi-bin` directory is indicative of executable files) or be coupled to a file type through its extension [34, 154]. Using the `flickr.com` example, the primary resource representation on that page has a URI[1] ending in ".jpg" and despite this extension in the URI, the type of the file returned is not guaranteed to be a JPEG-formatted image. Multiple URIs may identify the same resource (known as "URI aliasing" [85]). For example, the URI `https://matkelly.com/andMelissa` and the previous URI ending in ".jpg" return the same representation when dereferenced[2].

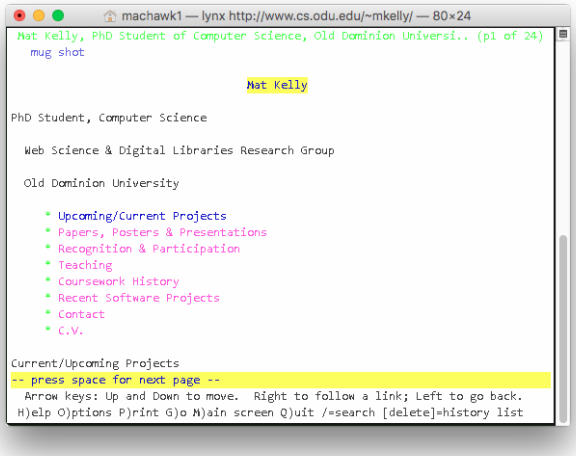
[1] `https://farm4.staticflickr.com/3160/2705987660_9aa5610f71_z_d.jpg`

[2] This can be juxtaposed to URIs "colliding" when the same URIs return drastically different representations per Section 1.2.



(a)

(b)



(c)

Fig. 14: Sample HTTP Response from Figure 12 when rendered with (a) a desktop graphical Web browser (Chrome), (b) a mobile Web browser (Brave [40]), and (c) a desktop terminal-based Web browser (Lynx [57]) as user-agents.

2.2 CONTENT ON THE WEB

Clients access the Web using a user-agent. A user-agent is frequently a Web browser (which need not be graphical, e.g. Lynx [57]), but the Web is also accessible with command-line or scripting tools, as with the "curl" user-agent in Figure 15 [30]. When a client requests a Web resource using a user-agent, the client often expects a Web page to be returned. Web pages typically consist of a text file, written in HyperText Markup Language (HTML), as well as the representations of other text-based resources like Cascading StyleSheets (CSS) and JavaScript (JS) files as well as potentially including embedded binary files like images. The representations of the other resources may be included inline within the HTML or, as more often occurs, included by their URI, which the user-agent then dereferences to render the Web page. The beginning of the entity body of an HTML document can be observed in Figure 12 starting with `<!DOCTYPE` (the document type declaration, used for parsing). Web browser user-agents (cf. command-line user-agents like curl) will attempt to parse the HTML file to be interpreted as a tree-based structure, called the Document Object Model (DOM) tree [75], based on the document type specified in the HTML representation. Figure 14 represents the same HTTP response when viewed in different browsers; Figure 14a using Google Chrome 58 for macOS, a graphical Web browser (user-agent); Figure 14b using Brave 1.0 for Android, also a graphical Web browser; and Figure 14c using Lynx 2.8 for macOS, a text-based Web browser.

```
$ curl -v https://matkelly.com/
> GET / HTTP/1.1
> Host: matkelly.com
> User-Agent: curl/7.54.0
> Accept: */*
```

Fig. 15: Sample HTTP Request (abbreviated for relevancy) to the URI-R `http://matkelly.com` using curl. Following this request, the server provides the response in Figure 12.

In additional to a hierarchical structure to inform the visual layout of a Web page, the DOM also provides language agnostic functions and attributes to manipulate the tree structure and represent Web Linking [130] via DOM element attributes. When parsing the DOM, a user-agent must perform subsequent HTTP requests to acquire

the representations of resources embedded on the HTML page. Both HTTP and HTML contain a mechanism for specifying inter-resource relations and both refer to the same registry, however, unlike the HTML definition for defining related resources [76] (e.g., the style.css file in Figure 12), relational resources in HTTP need not be format-specific [130].

The original Web that Tim Berners-Lee laid out has dramatically evolved. In a medium that initially consisted of static resources, other systems like databases were integrated to make the Web more useful. The URIs of some resources became more complex to generate, were only generated on-demand, or collided with multiple resources based on parameters beyond the URI (e.g., if a user is authenticated, per Section 1.2). Issues like these make comprehensive preservation more difficult to accomplish and evaluate.

JavaScript is a client-side programming language that allows creators of Web content to apply behavior to a Web page. This behavior can range from simply modifying the structure of a Web page to asynchronously dereferencing URIs. JavaScript may be embedded within HTML or reside in stand-alone files. With the advent of Web 2.0, Web pages became more interactive, particular in the realm of Asynchronous JavaScript and XML (AJAX) [68]. Many archival Web crawlers (e.g., Internet Archive's Heritrix [127]) do not support JavaScript execution, much less AJAX. Different approaches are taken to examine the secondary source files, once acquired, for URIs of additional resources with a recursive process in an attempt to dynamically and adaptably acquire the "deferred" representations [44] for all resources that are needed to replay a Web page.

2.3 CONTENT NEGOTIATION

Content negotiation on the Web is a means of serving different representations of a resource and can be accomplished using a variety of approaches. In this section we describe content negotiation in HTTP using `Accept-`, `Prefer`, `Cookies`, and `Features`. In Section 2.5 we discuss content negotiation in time in more detail due to it being primarily fundamental to our research.

HTTP 1.1 [61] defines the capabilities to perform multiple representations of one resource in a cache-friendly way using "Accept-" headers. Clients on the Web may engage in proactive content negotiation by sending HTTP request headers

like Accept-Charset, Accept-Encoding, and Accept-Language to specify acceptable character sets, encoding, and language (respectively) of the response [67]. For each specified value in the response header, a client may assign a corresponding quality value (from 0.000 to 1.000) to assign a relative weight to the preference. For example, a client sending the HTTP request header Accept-Encoding: compress;q=0.5, gzip;q=1.0 is indicating that they prefer the response to be "gzipped" and only secondarily, if the content cannot be gzipped, to return the content using the "compress" encoding. A quality value of 0 indicates that the preference is unacceptable [67]. For resources that have different representations based on the value in these headers sent, a "Vary" header in the HTTP response indicates that content negotiation on the specified dimensions is available. For example, Figure 16 shows an HTTP request being sent to the URI https://developer.mozilla.org with the HTTP request header of Accept-Language with a variety of values. In each instance, the resulting Location response header redirects the user to the URI of a representation that best aligns with the Accept-Language the client specified. In scenarios where the Accept-Language is unknown, unrecognized, or cannot be processed by the server (e.g., Accept-Language: odu), the server resolves the URI as it sees fit to best align with the request. While Accept- headers specify a preference, this preference may not be able to be met by the server.

Snell [160] introduced the Prefer HTTP request header to allow clients to specify a preference of behavior to be performed when a server performs content negotiation. Prior to the introduction of the header, HTTP offered no explicit means for a client to express a preference for optional aspects of a request beyond dimensions that have a corresponding Accept- header (e.g., Accept-Language, Accept-Charset). However, an implied expression of preference did exist in the Expect HTTP request header [67] but the requirements, as stated by Snell, were too strict for the expression of optional preferences. In comparison, the Prefer header contains extensible syntax with an expectation of additional preference values being valid to populate the header as defined in the future. Due to the dimensions of preference being potentially complex, Snell recommends not using Prefer for content negotiation. This issue may be mitigated by an optimization of the potential dimensionality as applied to the endpoint supporting Prefer and is explored in this research proposal. In Section 5.3.1 we utilize the Prefer header in the context of

additional arbitrary dimensions, where we anticipate that a server advertising the supported dimensions and values for those dimensions, the dimensionality ramifications of using `Prefer` will have minimal impact.

Barth [23] standardized the specification of HTTP State Management via Cookies, as was previously defined by Kristol and Montulli in two preceding specifications [112, 113]. Barth's approach at standardization was based on how the `Cookie` and `Set-Cookies` HTTP headers were actually used on the Web at the time. Cookies are a mechanism for an HTTP server to pass key-value pairs and associated metadata to a user-agent. When a user-agent accesses the server again, it can pass these values and infer an association with the data the client provided and potentially other information stored but not transferred on the server-side. A common use case for cookies is to store a session identifier with the client to simulate state as a client traverses a Web site. Cookies are widely supported in Web browsers, as the original specification by Kistol and Montuilli dates back to 1997 and is heavily utilized to provide a level of session persistence for user-agents. User settings and user preferences are frequently stored in client-side cookies and sent to HTTP servers to apply these setting upon requesting a resource. This loose correlation between the HTTP Prefer headers (`Prefer` and `Preference-Applied`) and Cookie headers (`Cookie` and `Set-Cookie`) may provide two approaches for client-side specification of personal and private Web archives to aggregators. While Prefer is more semantic, Cookies are more widely supported. Further, cookies are often opaque to the client and generated by servers while Prefer is intended to be initiated by the client. The merits of each approach are considered in Section 5.1.1.

Holtman and Mutz [77] standardized transparent content negotiation in HTTP, which allows multiple versions of the same resource to reside at the same URL. The intention of the specification was to be both scalable and interoperable for coexisting with other negotiation schemes. Each version of a negotiated resource is denoted as a "variant". The standard allows for extensibility to promote the "best" variant when an HTTP request is made. One intention of this specification was to remove error-prone and cache-unfriendly User-agent based negotiation, common in the Web when the spec was drafted in 1998. The specification also introduces the concept of a "transparently negotiable resource" that has multiple representations (variants) associated with it. In a related, more contemporary in-progress specification, Nottingham [133] is proposing the introduction of a `Variants` HTTP response header.

The introduction of this header would allow a server to enumerate the available variant representations. Much like the HTTP Prefer specification, a `Variant-Key` HTTP response header would accompany the `Variants` response header to indicate the representation variant of the response body.

```
$ curl -I https://developer.mozilla.org
HTTP/1.1 302 FOUND
...
Location: https://developer.mozilla.org/en-US/
Vary: Accept-Language
...

$ curl -I -H "Accept-Language: fr" https://developer.mozilla.org
HTTP/1.1 302 FOUND
...
Location: https://developer.mozilla.org/fr/
Vary: Accept-Language
...

$ curl -I -H "Accept-Language: odu" https://developer.mozilla.org
HTTP/1.1 302 FOUND
...
Location: https://developer.mozilla.org/en-US/
Vary: Accept-Language
...

$ curl -I -H "Accept-Language: en-CA" https://developer.mozilla.org
HTTP/1.1 302 FOUND
...
Location: https://developer.mozilla.org/en-US/
Vary: Accept-Language
...

$ curl -I -H "Accept-Language: es" https://developer.mozilla.org
HTTP/1.1 302 FOUND
...
Location: https://developer.mozilla.org/es/
Vary: Accept-Language
...
```

Fig. 16: `developer.mozilla.org` varies to which URI a user is directed based on the Accept-Language header supplied by the user. If none is sent, the site defaults to `en-US`. While some legal values cause the user to be directed to a different URI (`es` and `fr`), other valid values (`en-CA`) and invalid values (`odu`) simply resolve to the default.

2.4 WEB ARCHIVING

Web archives digitally preserve cultural heritage in the Web medium. The Web as a *medium* distinguishes it from simply being defined by the content it contains, as the Web is also as a container of the content, which allows it to be interpreted in a variety of ways [122] (e.g., different browsers with drastically different presentations of the content as in Figure 14). The Internet Archive (IA) and other institutional Web archives preserve content from the live Web for access by users at a later date. IA's archived Web content is publicly available and constitutes an example of a "public Web archive" (as described in Chapter 1).

2.4.1 WEB ARCHIVING IN PRACTICE

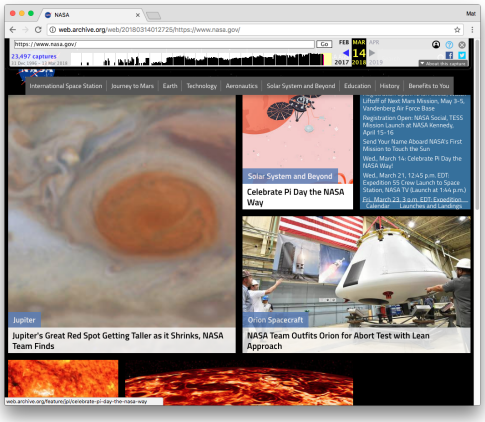
The National Digital Stewardship Alliance (NDSA) performed a survey [22] in 2016 (and previously in 2011 [128] and 2013 [21]) of organizations in the United States that preserve Web content. This most recent iteration of the survey highlighted relevant themes of collaborative Web archiving (mostly by all collaborators accessing a single service to provide URIs), access embargoes (used by fewer than 13% of respondents), and a survey of a wide range of tools used for organizational and personal Web archiving. The survey also highlighted questions asked about "data transfer" of Web archive data. The report states that most respondents (59%) were replicating their captures to local repositories, almost half (47%) to external repositories and 6% performing both operations. "For the first time", the report states, "trusting an external data capture service provider was the top reason for not replicating data to another repository". This is problematic, as we stated with our bearer examples in Chapter 1.

In 2011, Gomes et al. [72] performed a survey of Web archiving initiatives. They found that for the most part, Web archives are hosted in developed countries and run by small teams with a focus on acquisition and curation. They also discussed legal barriers and persistent issues with search mechanisms to enable access to these archives. Access to Web archives is a central theme is our work. Citing the importance of preserving content on the Web (with a particular example of born-digital photos as described in Figure 1), Gomes et al. highlighted other initiatives to evaluate the Web archiving landscape like one performed by the National Library of Australia

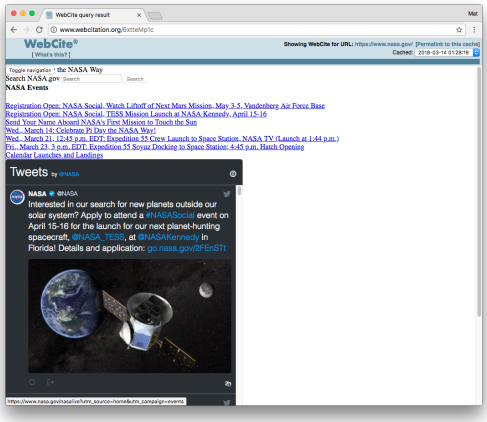
via the now ironically defunct Preserving Access to Digital Information (PADI) service [135] enumerating 17 major initiatives at the time and another study performed by the Joint Information Systems Committee (JISC) [55] that reviewed eight different initiatives. Each of these efforts to evaluate the landscape of Web archiving has been focused on public Web archives, as private or personal Web archives, while often smaller in number, may wish to not disclose their procedure and holdings for reasons that we hope to mitigate.

Other Web archives exist beyond Internet Archive. Some Web archives like the UK Web Archive[3] are scoped to only preserve certain parts of the Web – in this case, only "UK Web sites". Other Web archives like archive.is[4] and WebCite[5] [60] allow submission of Web pages to be archived by users in an on-demand basis using a Web form (Figure 17). A multitude of other archives exist, each with their own approach, scoping rules, and user submission allowances. Regardless of an institution-mandated crawl procedure or a system solely driven by user submissions, the existence of multiple Web archives provides a less centralized snapshot of the Web of the past.

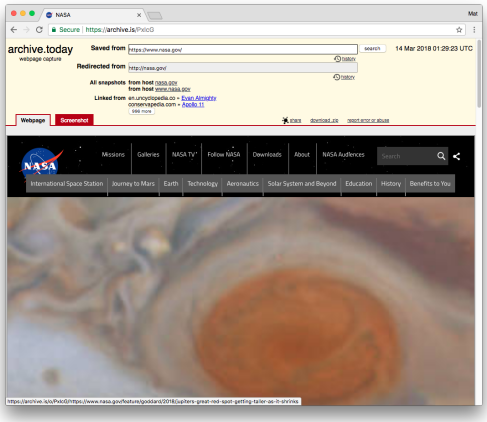
[3] `https://www.webarchive.org.uk/ukwa/`
[4] `https://archive.is/`
[5] `http://www.webcitation.org/`



(a) Internet Archive



(b) WebCite



(c) Archive.is

Fig. 17: `nasa.gov` as captured by three archives that allow immediate user-submitted preservation of URIs. Each page was captured within seconds of the other on March 13, 2018 despite the variance in results.

2.4.2 PRESERVING THE WEB

Two fundamental processes in Web archiving are the act of preserving content on the live Web and re-experiencing, or "replaying", the preserved content. Other processes exist both for accessibility of the content (e.g., indexing) and analysis of the content (e.g., metadata extraction, plaintext conversion).

One method of preserving the Web is to run an archival crawler that dereferences a URI, preserves the resource representation at the URI, extracts the URIs of embedded

resources and links, and repeats the process. These embedded URIs are stored in a "Frontier" until the process can be completed [157]. Heritrix [127] is an open-source, extensible, web-scale, archival-quality Web crawler created by the Internet Archive to preserve the live Web. Heritrix provides a variety of built-in options to allow users to leverage additional URI extraction methods as well as filters to limit the scope of crawls. After dereferencing a URI, Heritrix retains the entity body and HTTP headers of the transaction (Figure 12) and wraps the concatenated result in a record with metadata about the record (e.g., URI, time of capture) prepended onto the record. As Heritrix crawls additional URIs, these records are concatenated. This concatenation constitutes a "WARC file" where each record that was appended together is a "WARC record".

Other methods also exist to preserve the web, two of which are to preserve Web content as it is transferred from the server to client and on-demand archiving by URI. The first of the two methods are exhibited by tools like Webrecorder [143] and WARCreate [107]. With Webrecorder, a user visits the Web-based proxy at `https://webrecorder.io`, enters a URI, and "browses" the site and additional sites while content is preserved and replayable at the site. Users may also download their captures from this service. With the model performed by WARCreate, users install a browser extension that caches the content as they browse around. On the invocation of a procedure initiated by pressing a button within the extension's interface, a "WARC" file (discussed in Section 2.4.3) is generated and saved locally. While WARCreate does not provide a mechanism for replaying the captures, it does not require proxying all pages to a service outside of the user's machine to be preserved, thus facilitating more privacy at the expense of a seamless preserve-then-replay experience. These limitations are mitigated with a local replay system, as discussed with further details about purely client-side preservation in Section 4.1.

2.4.3 THE WEB ARCHIVE (WARC) FORMAT

Captures of the live Web by Heritrix and many other tools are often stored using the standard Web ARChive (WARC) format [82]. WARC files are made up of concatenated records. Some records describe the WARC itself (`warcinfo` and `metadata` WARC records, Figures 18a and 18b, respectively) while others contain the information and content of preserved live Web transactions (`response`

and `request` records, Figures 18c and 18d, respectively). Non-text-based representations of Web resources (e.g., the binary encoded content of an image shown in Figure 18d) are also concatenated alongside payloads containing textual content (e.g., Figure 18d). WARC `resource` records may also be used to archive other artifacts of a harvesting process inside a WARC file [82], related to but not necessarily served in the conventional HTTP request and response communication. `conversion` WARC records describe derivatives of other records after having performed some transformation on the original. An example of using a `conversion` record is to represent (in a `warc-response` record) a JPEG 2000 [159] formatted image (not viewable by many contemporary Web browsers) as a conventional JPEG in a `conversion` record with a field in the latter providing a reference to the former. `continuation` records also allow any record to be split amongst multiple other records, for instance, in cases where the desired file size of the WARC is exceeded (traditionally 1 gigabyte [69]). An example in using `continuation` records is to allow for a consistent file size between WARCs when created en-masse based on a potentially externally defined limitation in file size.

```
WARC/1.0
WARC-Type: warcinfo
WARC-Date: 2017-07-06T19:04:19Z
WARC-Filename: 20170706190419485.warc
WARC-Record-ID: <urn:uuid:0aa77218-cc3d-d266-df70-9595dba53ef7>
Content-Type: application/warc-fields
Content-Length: 463

software: WARCreate/0.2017.6.6 http://warcreate.com
format: WARC File Format 1.0
conformsTo: http://bibnum.bnf.fr/WARC/WARC_ISO_28500_version1_latestdraft.pdf
isPartOf:  basic
description:  Crawl initiated from the WARCreate Google Chrome extension
robots: ignore
http-header-user-agent:  Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_5)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36
http-header-from: warcreate@matkelly.com
```

(a) `warcinfo` record

```
WARC/1.0
WARC-Type: metadata
WARC-Target-URI: http://matkelly.com/frogsLeft
WARC-Date: 2017-07-06T19:04:19Z
WARC-Concurrent-To: <urn:uuid:dddc4ba2-c1e1-459b-8d0d-a98a20b87e96>
WARC-Record-ID: <urn:uuid:6fef2a49-a9ba-4b40-9f4a-5ca5db1fd5c6>
Content-Type: application/warc-fields
Content-Length: 71

outlink: http://matkelly.com/frogsLeftfroggies/frog.png E =EMBED_MISC
```

(b) `metadata` record

```
WARC/1.0
WARC-Type: request
WARC-Target-URI: http://matkelly.com/frogsLeft
WARC-Date: 2017-07-06T19:04:19Z
WARC-Concurrent-To: <urn:uuid:fd36167e-f96d-ad45-df14-de2f62b34dff>
WARC-Record-ID: <urn:uuid:d99b62e7-ebc1-23c9-89e7-2e3766ad5fa7>
Content-Type: application/http; msgtype=request
Content-Length: 364

GET /frogsLeft HTTP/1.1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_5)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8,de-DE;q=0.6
```

(c) `request` record

```
WARC/1.0
WARC-Type: response
WARC-Target-URI: http://matkelly.com/frogsLeft
WARC-Date: 2017-07-06T19:04:19Z
WARC-Record-ID: <urn:uuid:a48b1fe4-6d81-e038-f49a-5fef41ff1c67>
Content-Type: application/http; msgtype=response
Content-Length: 390

HTTP/1.1 200 OK
Date: Thu, 06 Jul 2017 19:04:17 GMT
Server: Apache
Vary: Accept-Encoding
Content-Length: 176
Keep-Alive: timeout=2, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<html><head></head><body>
<p>About 193 frogs left until the JCDL 2018 full paper deadline.</p>
<img src="froggies/frog.png" style="clear: both; width: 200px;">


</body></html>
```

(d) `response` record with HTML content

```
WARC/1.0
WARC-Type: response
WARC-Target-URI: http://matkelly.com/froggies/frog.png
WARC-Date: 2017-07-06T19:04:19Z
WARC-Record-ID: <urn:uuid:02467ae9-8da2-6ca4-a6cb-0f43b8a4b56c>
Content-Type: application/http; msgtype=response
Content-Length: 61797

HTTP/1.1 200 OK
Date: Thu, 06 Jul 2017 19:04:17 GMT
Server: Apache
Last-Modified: Mon, 11 Aug 2014 16:58:57 GMT
ETag: "f089-5005d7880a244"
Accept-Ranges: bytes
Content-Length: 61577
Content-Type: image/png

‰PNG
SUB
NUL NUL NUL
IHDRNUL NUL SOH |NUL NUL SOH -BS ACK NUL NUL NUL VT5Û¯NUL NUL NUL ENtEXtSoftware
NULAdobe ImageReadyqÉe<NUL NULð+IDATxÚìBEL |DC4eúÇ³³³ev³»I6=¤“žNULi„ŽDC4AŠˆ
ENQ ESC (SYN8ÅS,pŠw¢þåDC4ï„S±`ÁSTX
Š¨DC4é–”NULi„'DLEÔCzo›í;;³ÿ™
zz"$OLE!è|ùÌ'a³efçßüÞç}Þç%\.ETBDDDDDþøHÄ¯@DDDDC4 |DC1DC1DC1DC1DC1QðEDDDDD
ÁETBù]8[R£NAK~fNAKVENQˆß†ˆˆÈå NAK¿,îÍDC3…C3sÎ½RÚbFSê´9(BELBSEOTè=²ƒý½v
EOTúx¦îUS»Yü–DDD:STX!fétOŽæ”&íËÈùæà‘ÜX³ÕSOÈ)€åNUL’ä²P*e
årDEOTzcÏ€¸Ç&èµLüÖDDDDÁ¿ÎØu4oÊ‡ßìY×P×STX(xí–žïˆDC1EOTà>_üOÖÉÿ¸UÈ(Üyc¿åSIÝ<
xVw9†ÅÔ*Šq8´SOÆiNAKvYiP4ÆE…SUBÃ³+""
¾Èy¾YsbþŠÍ‡ETBEMÛ,€„Ws’ETBôVT"þO?=ÎqPËe¸ù†Þ«US¹mø´Î~æ™â*½Ád‹”’DC2‹RN5&÷FF®
¹¢ÞÉ&ì”ªšši EOT\o³|o„DC4z%Å<ÿô}{GSDC1Ï²ˆˆ(
øzNæW„¬Ü^v"¯FFà]{ÇÏ"¯þFFÃwEOTHÏ»ÿ¦iãBEL'®¾ÔKòËê¨ü’ê™é§
ßol1¢ÁÅh…\&…·‡DC2~^å=|=w$FENQ/GSOLE–Ý™c8x1ÃÐêªš‘DC2’xEÅ÷¢BS~ß~lcüïóbd£WÃõFF
oDC4Ï¶ˆˆ(øj>ÛxðýÕESCÔg;EOTgÏ‹7:sjEOTÇì`DC1êï…¿Ïš¤Š
```

(e) `response` record with binary content

Fig. 18: WARC files consist of concatenated records representative of a live Web capture (18c, 18d, and 18e), metadata about the WARC (18b), derivative data based on the capture (18a and 18b), and additional supplementary content for the capture.

2.4.4 OTHER WEB ARCHIVE FORMATS

In addition to the WARC format (Section 2.4.3), other supplementary formats in the Web archive workflow allow the contents in the captures to be more accessible. In this section we will discuss the CDX and CDXJ formats and how they relate to the framework we describe in this research.

CDX

The CDX file format [83] is a *de facto* standard format used by Internet Archive and OpenWayback that serves as an index to WARC files and to associate fundamental metadata about the capture based on the WARC contents. This metadata is limited to space-delimited fields like the SURTed (Sort-friendly URI Reordering Transform) [158] URI-R, datetime, status code, MIME-type, etc. CDX records within CDX files are delimited by line breaks between records and the fields within a record are delimited by a space character. Figure 19 shows an example of a CDX record of a capture of `https://matkelly.com` at January 12, 2016 9:49am GMT (represented by the 14-digit datetime, 20160112094927).

The initial field of a CDX record is the SURTed URI-R, which represents the canonicalized version of the URI when preserved from the live Web. Canonicalization allows after-the-fact clustering of URIs that likely reference the same resource [98, 97]. For example, the "www" subdomain is often used on the live Web to represent the same content as the version of the representation without this subdomain. In this case, it is likely that the content at `http://matkelly.com` and `http://www.matkelly.com/` is the same and thus the canonicalization method of coalescing the two URI-Rs is often performed in generating CDX entries when indexing a WARC containing captures of each of these URI-Rs. Other canonicalization rules may be applied like scheme-level canonicalization of the previous URIs with `https://www.matkelly.com` and URIs that include a path (like `http://matkelly.com/index.html` and `http://www.matkelly.com/default.asp`) that often resolved to a URI without the path. In practice, URIs with well-known subdomains (e.g., www), a slight difference in scheme (e.g., http(s)), and common paths (e.g., index.html) are all canonicalized into the same resulting string within a CDX record.

```
com,matkelly)/ 20160112094927 http://matkelly.com/
↪ text/html 200 I6PPTO3TGZG4X7RZQHADKGC45QXAEODR 5243 - -
↪ 656 900 myCaptures.warc.gz
```

Fig. 19: An example CDX index record maps a capture of `matkelly.com.com` to a WARC file named myCaptures.warc.gz. The entirety of a CDX record resides on a single line. Line breaks are shown here for clarity.

CDXJ

CDXJ is an extension of CDX that contains a JSON block with a memento's attributes. Much like CDX, CDXJ records are line delimited. Fields within a CDXJ record are space-delimited with the final field consisting of a JSON block (encapsulated with curly braces, i.e., {}). While its relevance to Memento is discussed in upcoming Section 2.5, CDXJ provides semantics of WARC indexes using an extensible approach facilitates by the JSON block. The implicit expectation of using the JSON block for attributes instead of a rigid set of fields is that archives may supply additional attributes to the index for external use without breaking the expectation of ordering by different tools. Parsing values from CDX will likely be based on the fields' ordering for semantics whereas object-based parsing semantic attribute retrieval prevents parsing implementations from breaking as new attributes are added, so long as the base attributes for a memento are expressed.

```
20160112094927 {"uri": "http://matkelly.com",
↪ "rel": "memento", "datetime": "Tue, 12 Jan 09:49:27 GMT"}
```

Fig. 20: The CDXJ record for the same CDX entry Figure 19 as expressed in a CDXJ-formatted TimeMap served from MemGator. The line break is added for clarity, as a single CDXJ record resides on a single line.

2.4.5 REPLAYING WEB ARCHIVES

WARC files cannot be natively interpreted by Web browsers. To re-experience the contents of a WARC, the contents of the WARC records must be extracted and re-assembled to replicate the original process of the live Web page being assembled, a procedure called "replay". The Internet Archive's Wayback Machine was created

to read both WARC files and archives of WARC's predecessor, the ARC format [51], and replay the contents through a Web browser. With the scale of a Web archive's holdings being large (over 658 billion web objects as of July 2018 [43]), the contents requested for a URI or an embedded resource on a page may exist in multiple WARC files. Internet Archive's *Wayback Machine* [165], its open-source derivative project *OpenWayback* [79], and *pywb* [111] are examples of replay systems that are able to perform this re-assembling of archived Web pages at scale.

The replay process requires an indexing procedure of the WARC files to efficiently map requests for a URI at a datetime to a certain location in a particular WARC file. The procedure produces index files, often stored in the CDX format in practice, but the procedure for replay is generally the same regardless of the formats used. When a client requests a URI at a datetime, the replay system refers to its collection of indexes to obtain the source and offset (location in the source) of the payload to be returned to the client. When this payload (e.g., an HTML page) is interpreted by a user-agent, the agent (per its conventional functionality) parses the payload (e.g., into a DOM tree) and requests the embedded resources contained within the payload as if on the live Web. Replay engines will often rewrite the URI of the embedded resource so as to point to identifiers of archived resources within the archive itself instead of pointing to the live Web.

2.5 MEMENTO

In Section 2.4 we discussed multiple organizations' efforts to preserve the Web. With both these services and individuals' Web archives coming and going over time, it is useful to be able to query multiple archives at once. Doing so gives a more temporally comprehensive picture of the Web as it once existed. Memento [169] is a framework that adds the dimension of time to the Web - a critical characteristic for Web archive access by providing a universal versioning system. Memento terminology is used throughout this research. A large portion of public Web archives (including IA) support Memento. Memento specifies the term *URI-M* as a URI of an archived representation of a live Web resource and *URI-R* as a URI for a live Web resource (Figure 21).

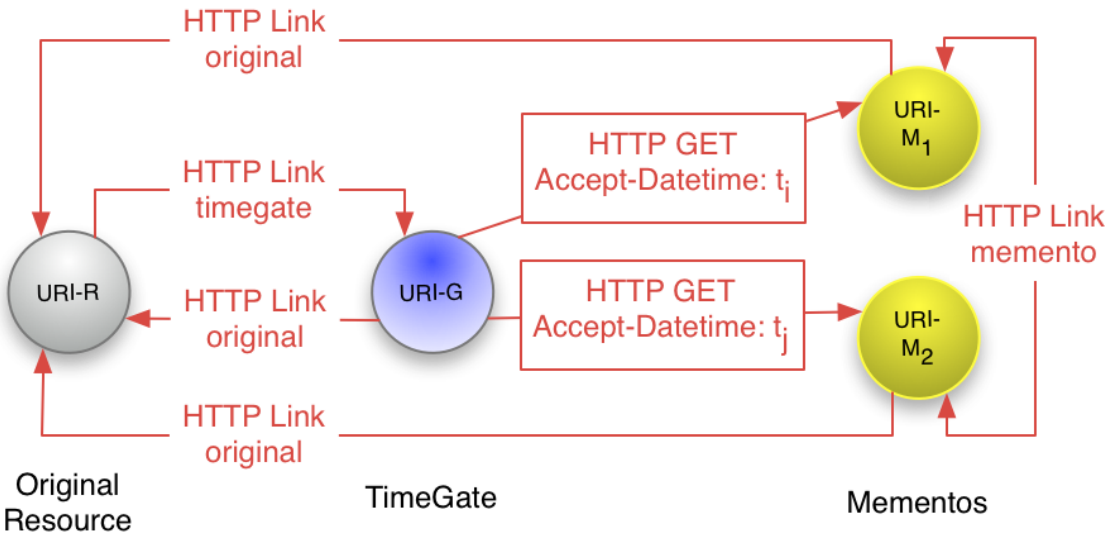


Fig. 21: Memento provides the ability to associate live Web and archived Web captures (at URI-Rs and URI-Ms, respectively), relations between URI-Ms for the same URI-R, and negotiation of resolving a datetime closest to one specified in an HTTP Accept-Datetime header using a TimeGate (at URI-G) [2].

Memento provides a mechanism for accessing the past Web using date-based content negotiation via the Accept-Datetime HTTP request header. Content negotiation in a dimension where the variants are countably infinite (i.e., time), as compared to conventionally finite variants (e.g., like `Content-language` [17]), requires additional HTTP entities to handle the negotiation. An additional Memento entity called a TimeGate, identified by a URI-G, handles the date-based requests for a URI-R. A client may provide this header at the time of request along with a datetime value [39] to a TimeGate with the expectation that the recipient Memento-compliant Web archive will resolve the datetime to return the URI-M closest to the value of the `Accept-Datetime` header provided (Figure 21). To distinguish live Web from archival Web captures, Memento enables the HTTP `Memento-Datetime` response header. Figure 22 shows an example with a client sending the `Accept-Datetime` HTTP request header to a TimeGate at URI-G and receiving the `Memento-Datetime` HTTP response header when requesting a capture for `http://matkelly.com` at January 9, 2007 at midnight GMT. The requested TimeGate responds with an HTTP 302 (Found) response and directs the client a different URI using the HTTP `Location` response header [67]. When the user-agent sends a subsequent request for the URI (Figure 23) to which they

were redirected (redirects are often performed transparently for the user by the agent), the resource representation returned reports an HTTP 200 status code and a `Memento-Datetime` HTTP response header. The latter is indicative of the second URI being a URI-M, i.e., the representation returned is a memento. Were the user redirected to a URI that did not include a Memento-Datetime response header (e.g., if the TimeGate sent the user to a live Web URI, another URI-M without a `Memento-Datetime`, or a memento at a non-Memento-compliant archive), the returning representation would not be indicative of an archival capture (memento).

```
curl -v -H "Accept-Datetime: Tue, 9 Jan 2007 00:00:00 GMT"
↪ http://web.archive.org/web/http://matkelly.com
*   Trying 207.241.225.186...
* Connected to web.archive.org (207.241.225.186) port 80 (#0)
> GET /web/http://matkelly.com HTTP/1.1
> Host: web.archive.org
> User-Agent: curl/7.54.0
> Accept: */*
> Accept-Datetime: Tue, 9 Jan 2007 00:00:00 GMT
>
< HTTP/1.1 302 FOUND
< Date: Tue, 27 Mar 2018 02:09:07 GMT
< Content-Type: text/plain; charset=utf-8
< Content-Length: 32
< Connection: keep-alive
< Location:
↪ http://web.archive.org/web/20060717055501/http://www.matkelly.com:80/
< Vary: accept-datetime
< Link: <http://matkelly.com>; rel="original",
↪ <http://web.archive.org/web/20060717055501/http://www.matkelly.com:80/>;
↪ rel="memento"; datetime="Mon, 17 Jul 2006 05:55:01 GMT",
↪ <http://web.archive.org/web/timemap/link/http://matkelly.com>;
↪ rel="timemap"; type="application/link-format"
<
found capture at 20060717055501
```

Fig. 22: Datetime negotiation using Memento consists of a user requesting a URI-M for a TimeGate with an Accept-Datetime header value in the HTTP request. Upon receiving the request, the TimeGate returns the closest URI-M to the requested date in an HTTP response with an HTTP redirect.

```
$ curl -v
↪ http://web.archive.org/web/20060717055501/http://www.matkelly.com:80/
> GET /web/20060717055501/http://www.matkelly.com:80/ HTTP/1.1
> Host: web.archive.org
> User-Agent: curl/7.54.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Tue, 27 Mar 2018 02:10:27 GMT
< Content-Type: text/html; charset=iso-8859-1
< Content-Length: 2735
< X-Archive-Orig-date: Mon, 17 Jul 2006 05:55:01 GMT
< X-Archive-Orig-connection: close
< X-Archive-Orig-server: Apache/1.3.33 (Unix) mod_throttle/3.1.2
↪ DAV/1.0.3 mod_fastcgi/2.4.2 mod_gzip/1.3.26.1a PHP/4.4.2
↪ mod_ssl/2.8.22 OpenSSL/0.9.7e
< Memento-Datetime: Mon, 17 Jul 2006 05:55:01 GMT
< Link: <http://www.matkelly.com:80/>; rel="original",
↪ <http://web.archive.org/web/timemap/link/http://www.matkelly.com:80/>;
↪ rel="timemap"; type="application/link-format",
↪ <http://web.archive.org/web/http://www.matkelly.com:80/>;
↪ rel="timegate",
↪ <http://web.archive.org/web/20060514123511/http://www.matkelly.com:80/>;
↪ rel="first memento"; datetime="Sun, 14 May 2006 12:35:11 GMT",
↪ <http://web.archive.org/web/20060711174742/http://www.matkelly.com:80/>;
↪ rel="prev memento"; datetime="Tue, 11 Jul 2006 17:47:42 GMT",
↪ <http://web.archive.org/web/20060717055501/http://www.matkelly.com:80/>;
↪ rel="memento"; datetime="Mon, 17 Jul 2006 05:55:01 GMT",
↪ <http://web.archive.org/web/20090505173357/http://matkelly.com:80/>;
↪ rel="next memento"; datetime="Tue, 05 May 2009 17:33:57 GMT",
↪ <http://web.archive.org/web/20180319141920/http://matkelly.com/>;
↪ rel="last memento"; datetime="Mon, 19 Mar 2018 14:19:20 GMT"
<
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
...
```

Fig. 23: When a user-agent receives a redirect (Figure 22) when dereferencing the URI-G, the URI-M returned from the TimeGate is subsequently requested and the HTTP response returned to the user-agent for the user.

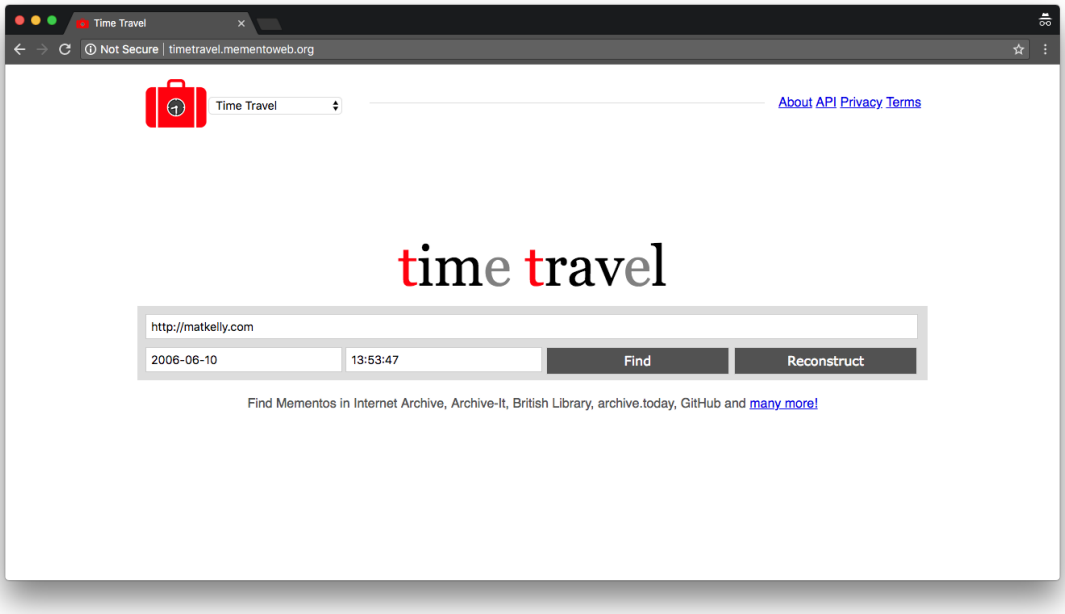
```
$ curl -v https://memgator.cs.odu.edu/timemap/link/https://matkelly.com
> GET /timemap/link/https://matkelly.com HTTP/1.1
> Host: memgator.cs.odu.edu
> User-Agent: curl/7.54.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Content-Type: application/link-format
< X-Generator: MemGator:1.0-rc7
<
<http://matkelly.com>; rel="original",
<http://memgator.cs.odu.edu/timemap/link/http://matkelly.com>;
↪ rel="self"; type="application/link-format",
<http://web.archive.org/web/20070302064530/http://www.matkelly.com/>;
↪ rel="memento"; datetime="Fri, 02 Mar 2000 06:45:30 GMT",
<http://arquivo.pt/wayback/20070407215431/http://www.matkelly.com/>;
↪ rel="memento"; datetime="Sat, 07 Apr 2000 21:54:31 GMT",
<http://archive.is/20160302231212/https://matkelly.com/>;
↪ rel="memento"; datetime="Wed, 02 Mar 2016 23:12:12 GMT",
<http://wayback.archive-it.org/all/20160304000513/matkelly.com/>;
↪ rel="memento"; datetime="Fri, 04 Mar 2016 00:05:13 GMT",
<http://web.archive.org/web/20160304031820/http://www.matkelly.com/>;
↪ rel="memento"; datetime="Fri, 04 Mar 2016 03:18:20 GMT",
<http://memgator.cs.odu.edu/timemap/link/http://matkelly.com>;
↪ rel="timemap"; type="application/link-format",
<http://memgator.cs.odu.edu/timemap/json/http://matkelly.com>;
↪ rel="timemap"; type="application/json",
<http://memgator.cs.odu.edu/timemap/cdxj/http://matkelly.com>;
↪ rel="timemap"; type="application/cdxj+ors",
<http://memgator.cs.odu.edu/timegate/http://matkelly.com>;
↪ rel="timegate"
```

Fig. 24: An abbreviated TimeMap (prepended with the verbose HTTP request and response headers) from a Memento aggregator shows URI-Ms for the URI-R `matkelly.com` from Internet Archive (`archive.org`), Archive-It (`archive-it.org`), Portuguese Web Archive (`arquivo.pt`), and Archive.is (`archive.is`).

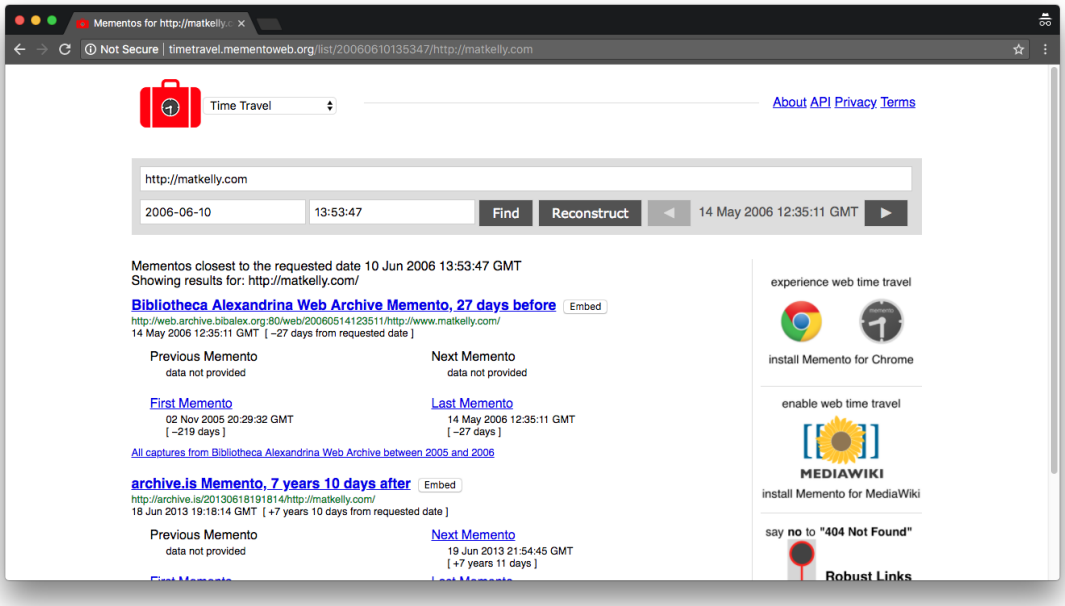
A Memento aggregator is an entity that acts as an endpoint for querying and combining the identifiers (URI-Ms) for archived representations (mementos) from multiple Web archives. A Memento aggregator provides access to the chronologically

ordered results of the mementos (accessible by dereferencing a URI-M) of content that reside in Web archives (mementos) that constitute prior representations (and were once accessible at a URI-R). The listing of the mementos returned from a Web archive or from a Memento aggregator is provided as a TimeMap. For example, Web archives A, B, and C contain URI-Ms for a URI-R R, labeled as $\{A\}$, $\{B\}$, and $\{C\}$, respectively. If queried individually, each respective archive would return a TimeMap containing its mementos for a URI-R. When a Memento aggregator configured to request the URI-Ms from these three archives receives a request with R as a parameter, the aggregator would return a TimeMap containing the URI-Ms $\{A, B, C\}$, along with other information like the URI-R, URI of the TimeMap itself (defined as *URI-T*), etc. Memento aggregators may also provide identifiers for TimeGates and TimeMaps from multiple archives. Figure 24 shows an abbreviated Link-formatted [130] TimeMap containing URI-Ms from multiple archives (e.g., `arquivo.pt`, `archive.is`, and `web.archive.org`) as returned from a Memento aggregator for `matkelly.com`. The TimeMap also contains URI-Ts for TimeMaps in two other formats (JSON [41] and CDXJ [5, 10], the latter discussed in Section 2.4.4) and other identifiers for other Memento entities, described below.

A deployed implementation of a Memento TimeGate and aggregator resides at `mementoweb.org`'s Time Travel service. When accessing this Web page, a user is presented with an interface (Figure 25a) to specify a URI-R and datetime for submission to the aggregator. The aggregator receives the user's input and provides a second Web page with the results (Figure 25b) including temporal proximity of the nearest memento at the URI-R of the date and time specified and a by-archive breakdown of the results, also including the temporal proximity.



(a) Homepage of Time Travel service at `mementoweb.org`



(b) Results from Time Travel for a request for a Memento nearest June 10, 2006 13:53:47 for `matkelly.com`

Fig. 25: The Time Travel service at `mementoweb.org` provides a user-friendly interface to a Memento TimeGate and aggregator.

CDXJ in Memento

In addition to serving as a richer index for WARC files beyond CDX (Section 2.4.4), the CDXJ format has also been adapted as an alternative format to Link [130] for Memento TimeMaps. Figure 26 shows corresponding Link and CDXJ TimeMaps for `http://matkelly.com`. This figure highlights the parallels including the representation of the URI-R, URI-G, URI-Ts, and URI-Ms, as highlighted. Each TimeMap format variant here also provides a URI-T to the other variant and an additional TimeMap in the JSON format (not pictured), which has similar parallels. The attributes about a URI-M in the Link TimeMap are limited to those defined in the Memento and Web Linking specifications, where the attributes for each URI-M in the CDXJ-formatted TimeMap may be extended within the JSON block of each record to be more descriptive about the respective URI-M in the context of a TimeMap. This parallel between the extensibility that CDXJ provides beyond the basis standards (CDX indexes and Link TimeMaps) is subtle yet powerful. In the CDX use case, the de facto standard provides no explicit semantics and the implicit semantics depend on the order of the values provided for each record. In the Link use case for CDXJ, the degree of descriptiveness that is syntactically available while still adhering to the reference specifications (Memento and Link) is limited, preventing descriptors that may be solely useful to describing mementos (cf. the applicability to the Web in general of Web Linking) from being both expressed and conforming.

```
!context ["http://tools.ietf.org/html/rfc7089"]
!id "uri":  "http://localhost:1208/timemap/cdxj/http://matkelly.com"
!keys ["memento_datetime_YYYYMMDDhhmmss"]
!meta "original_uri":  "http://matkelly.com"
!meta "timegate_uri":  "http://localhost:1208/timegate/http://matkelly.com"
!meta "timemap_uri":  "link_format":
"http://localhost:1208/timemap/link/http://matkelly.com", "json_format":
"http://localhost:1208/timemap/json/http://matkelly.com", "cdxj_format":
"http://localhost:1208/timemap/cdxj/http://matkelly.com"
20060514123511 "uri":
"http://web.archive.org/web/20060514123511/http://www.matkelly.com:80/", "rel":
"first memento", "datetime":  "Sun, 14 May 2006 12:35:11 GMT"
20060516213852 "uri":
"http://web.archive.org/web/20060516213852/http://www.matkelly.com/", "rel":
"memento", "datetime":  "Tue, 16 May 2006 21:38:52 GMT"
...
20180128152125 "uri":
"http://web.archive.org/web/20180128152125/http://matkelly.com", "rel":  "memento",
"datetime":  "Sun, 28 Jan 2018 15:21:25 GMT"
20180319141920 "uri":
"http://web.archive.org/web/20180319141920/http://matkelly.com/", "rel":  "last
memento", "datetime":  "Mon, 19 Mar 2018 14:19:20 GMT"
```

```
<http://matkelly.com>; rel="original",
<http://localhost:1208/timemap/link/http://matkelly.com>; rel="self";
type="application/link-format",
<http://web.archive.org/web/20060514123511/http://www.matkelly.com:80/>; rel="first
memento"; datetime="Sun, 14 May 2006 12:35:11 GMT",
<http://web.archive.org/web/20060516213852/http://www.matkelly.com/>; rel="memento";
datetime="Tue, 16 May 2006 21:38:52 GMT",
...
<http://web.archive.org/web/20180128152125/http://matkelly.com>; rel="memento";
datetime="Sun, 28 Jan 2018 15:21:25 GMT",
<http://web.archive.org/web/20180319141920/http://matkelly.com/>; rel="last memento";
datetime="Mon, 19 Mar 2018 14:19:20 GMT",
<http://localhost:1208/timemap/link/http://matkelly.com>; rel="timemap";
type="application/link-format",
<http://localhost:1208/timemap/json/http://matkelly.com>; rel="timemap";
type="application/json",
<http://localhost:1208/timemap/cdxj/http://matkelly.com>; rel="timemap";
type="application/cdxj+ors",
<http://localhost:1208/timegate/http://matkelly.com>; rel="timegate"
```

Fig. 26: A CDXJ TimeMap (top) represents the same content as a Link TimeMap (bottom) including the URI-R (`http://matkelly.com`, highlighted in red), URI-G (blue), other URI-Ts (green), and URI-Ms (brown) with identical relations (note similarity of the corresponding rel attributes).

2.6 ACCESS CONTROL

Accessing content on the live Web often requires some form of access control, often implemented as the service hosting the content requiring a user to supply credentials or authenticate through another means. Content behind authentication is often inherently personal and/or private. When this content is preserved, it is decoupled from the original authentication mechanism on replay. To account for this, Web archival replay and access through other methods requires some form of access control to regulate potentially private information being served. The remainder of this section focuses on reusing a "live Web" standard authentication mechanism as will be latter applied to the archived Web.

OAuth 2.0 [74] is an open standard for providing authorization for resources on the Web through a means of secure delegation of access without loss of access control. OAuth 2.0 defines four roles of entities in its framework: a resource owner, a resource server, a client, and an authorization server. The model described by the specification (Figure 27) entails a client requesting authorization from a resource owner, passing this grant to an authorization server to obtain a token, then using this token for requests for resources from a resource server. An access token is a string representing an authorization issued to the client entailing attributes of access like duration, scope, etc. In developing the framework for this research, we investigate using OAuth 2.0 as implemented on the live Web to establish authorization and regulate access to private archives using OAuth's bearer tokenization model [86]. Regulating access beyond a simple "accept or deny" scheme requires an extensible system to accommodate private Web archives' need to tailor access to the resources.

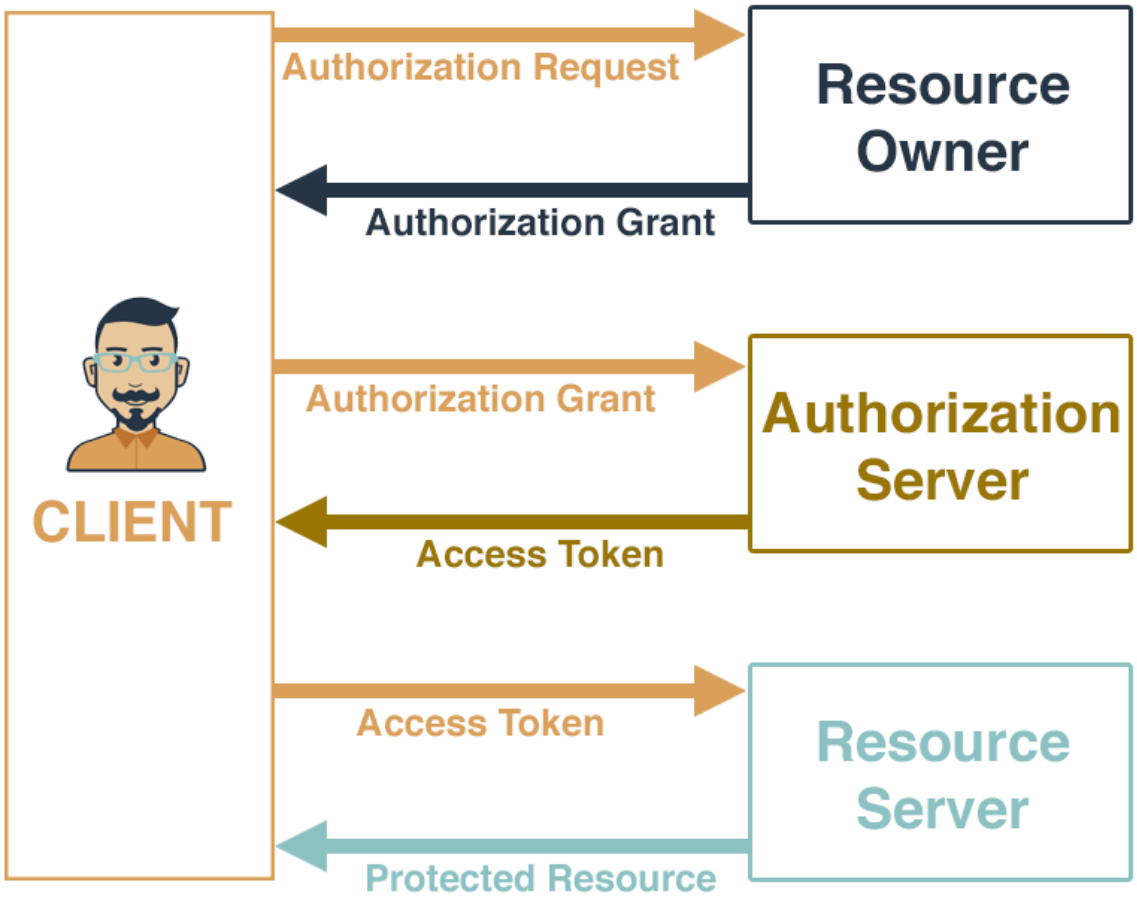


Fig. 27: The OAuth 2.0 abstract protocol flow decouples the resource owner, resource server, and authorization client using a token-based system for access persistence.

This common authentication model is often seen from a client's perspective with the details hidden. For example, when viewing a blog post on which a user wishes to leave a comment, there will often be the option to authenticate using another service, e.g., a user may use their `facebook.com` credentials to comment. Rather than the blog (the resource owner) being required to authenticate the user's Facebook account, when the user performs an "authorization grant", i.e., requests permission to comment, the blog provides the authorization grant mechanism of authenticating through Facebook. The user can then use this grant to authenticate to Facebook's "authorization server". Upon successful authorization, Facebook returns an access token to the user. The user can then provide this token when commenting to associate access to the "protected resource", here the ability to comment, with subsequent requests. This sort of token persistent prevent the user needing to authenticate with each post, prevents the blog from needing to maintain authentication, and allows

the user or blog to de-authenticate the access by disavowing the token. Upon a token being disavowed, when verified with the authorization server on subsequent comment attempt, a response will indicate this and instruct the resource server to prevent access to the protected resource.

In the context of Web archiving, no such authentication procedure is typically performed when accessing Web archives. Archival crawlers perform a capture of the representation of the resource without enforcing subsequent access restrictions, as the authorization server's functionality is not preserved. In this work we adapt the OAuth 2.0 procedure to regulate access in the context of the aggregation of public and private Web archives.

2.7 SUMMARY

In this chapter, we performed a high-level review of the fundamentals of the Web and Web Archiving. We then outlined the foundational technologies and advancements in Web archiving. Finally, we gave an overview of fundamentals of access and security that are relevant and a prerequisite for exploring the research described in this work. Each of these sub-topics, combined together, serves as the basis on which we build the framework for aggregating private and public Web archives.

CHAPTER 3

RELATED WORK

In this chapter we discuss previous research about Web archiving, HTTP, and security. Section 3.1 discusses research related to Memento and HTTP mechanics, which we will build upon while exploring RQ5. Section 3.2 discusses relevant work on privacy and security that will guide us in determining appropriate means of access control of private Web archive contents (RQ6). As Web archives proliferate, migration is key in assuring their posterity. Section 3.3 describes research performed in propagation and sharing of Web archives both within an organization and on a smaller, personal scale. Section 3.4 describes work related to distinguishing and finding the similarities and potential reuse between public, private, and personal Web archives. This final section of this chapter will assist us on the aggregation aspect between the three, particularly on how to distinguish them when additional considerations are needed (RQ4, RQ5, and RQ6).

3.1 MEMENTO AND HTTP MECHANICS

This section highlights relevant research exploring aspects of Memento beyond the original Memento specification and the fundamental research described in Section 2.5.

3.1.1 MEMENTO TIMEMAPS

In previous work [98, 97], we performed a deep dive into the identifier for mementos (URI-Ms), highlighting the fallacy of relying solely on a TimeMap to determine a count for the number of Mementos available. In an investigation primarily into `google.com`, we found that 84.9% of the URI-Ms in the TimeMap returned an HTTP redirect when dereferenced. This indicates that the URI-Ms must be requested to obtain a true count of the number of representations a TimeMap represents. In this work we sampled from the Internet Archive's Memento endpoint, the CDX Server endpoint, and the explicit count of captures presented in the Wayback Machine interface. The number of mementos available for a URI-R as conveyed by each of these methods varied depending on which source and method of counting was used.

3.1.2 MEMENTO AGGREGATION

AlSum et al. [16] studied the routing of URI lookups to Web archives where the conventional model for aggregators is to broadcast the request to all archives as configured, which is inefficient. Even with a listing of URI-Ms from a TimeMap, the URI-Ms whose content is accessible varies with the accessibility of the target archive, which varies with time as archives come on and offline [155]. The current management of adding and removing Memento-compatible archives to the Memento aggregator software is a manual process with no subscription-like model nor an API for manipulating the set of archives included in-place.

Bornand et al. [38] highlighted a problem for Memento aggregators where as the progressive number of archives aggregated increases, so does the response time and computation costs of the aggregator. Using cached queries to the archives, they were able to develop a binary classifier to determine whether a particular archive ought to be queried based on the request from the client. Using their findings, they were able to decrease the average number of requests by 77% and reduce the response time by 42%. Bornand et al. also emphasized the necessity for aggregators to implement selective polling of supported archives with practical examples of recent services deployed that indirectly increased traffic to the archives via the aggregator and caused a dramatic increase in response time. In a production environment for their aggregator they found that just over 82% of the URI-Rs covered by their aggregator's configuration have mementos in only 0, 1, or 2 of the supported archives (inclusive of archives with which they interface by proxy).

Alam et al. [11, 10] profiled Web archives using sampling (i.e., examining the archives' contents cf. Bornand et al. examining TimeMap responses) to mitigate the need of an archive to explicitly update a representation of its holdings. Using a profile, a Memento aggregator is able to more efficiently route requests for mementos for a URI-R based on the relevancy of the URI-R to the archives' respective captures. Their sampling method was accomplished through a crawling procedure, which may require adjustment for archives with a more complex access scheme as proposed in this work. However, because a personal or private Web archive's holdings are likely much smaller than most institutions, the rate of unnecessary querying to personal archives when aggregated would likely be much higher. The work of Alam et al. will help inform the design of the modified aggregator in this proposal to allow an aggregator to better advertise the relevancy of potential queries where applicable and

allowed per access restrictions. Because exposing the metadata of an archive also has ramifications, a special case may be needed when indicating the presence of captures via the derivative profiling attribute. This will need to be expressed in a way that does not unnecessarily expose a private Web archive's holdings.

Brunelle and Nelson [49] studied archives so as to recommend caching policies for Memento aggregators, a process that aggregators use to optimize the temporally expensive operation of querying and aggregating the URI-Ms from multiple archives. As a contribution of that work, they found that TimeMaps are not necessarily monotonically increasing in size. When disks hosting the contents of archives die, a subset of archives with URI-Ms in a TimeMap come on- and off-line, and because of a slew of other circumstances (both engineering and policy based), the set of mementos identified in a TimeMap may change. The proliferation of personal Web archives amplifies the importance of our work in this proposal, as Web archives hosted by individuals are likely to be less consistent with uptime and reliability compared to their institutional Web archive counterparts.

Rosenthal [147] highlighted further issues with the then-current state of Memento aggregators, with particular relevance to his notes on aggregator scalability. Memento provides no structure to represent and differentiate mementos originating from private Web archives with those from public Web archives. In this proposal, we examine methods to strategically identify the different sorts of captures. Further, we define methods to appropriately handle the captures based on the attributes for the identifiers. Rosenthal [146] emphasized that temporal order may not be optimal for TimeMaps returned from Memento aggregators. He stated that aggregators need to develop ways of estimating usefulness of preserved content and conveying these estimates to readers. In a different work, Rosenthal [145] described the behavior of aggregators returning "Soft 403s" consisting of captures of login pages when the user likely expected content shown that was originally behind authentication. Rosenthal [145] also described a "hints list" that an aggregator might provide based on its own experience of requesting content from archives. In this work, Rosenthal also alluded to a hypothetical mechanism of the aggregator filtering content like login pages from the results and redirecting a user to a version of the TimeMap containing only captures that are not a login page.

Memento currently allows URIs for mementos, TimeMaps, and TimeGates (URI-Ms, URI-Ts, and URI-Gs, respectively) to be aggregated by a Memento aggregator and

returned to a user sending requests to a Memento endpoint. Specification of the set of Memento-compliant archives (public or otherwise) are included at the disposal of maintainer of these aggregators. Alam and Nelson's MemGator [9] allows anyone to deploy their own Memento aggregator and to include a set of Web archives as defined via a configuration file prior to launching the application. We extended the software in this work to account for the privacy and access control aspects beyond the considerations that the Memento framework addresses.

3.1.3 TYPES OF MEMENTOS

Jones et al. [87] discussed obtaining the "raw mementos" consisting of un-rewritten links in captures in a systematic way using the HTTP Link response header. By utilizing the HTTP Prefer request header [160], a user would be able to obtain a version of the memento as it appeared at the time of capture instead of a version with relative links rewritten by the archive to point back within the archive and not the live Web. An archive, in response and to confirm compliance with the request, would return the memento with the HTTP Preference-Applied response header along with the requested original version of the memento.

Van de Sompel et al. [170] highlighted that the Prefer header could be used by Web archives to allow clients to specify a request for the unaltered or un-rewritten content. Rosenthal [148] echoed Van de Sompel et al. by suggesting a list of transformations (screenshot, altered-dom, etc.) for a memento via a new HTTP header. To resolve URI-Ms of embedded resources, a replay system will often perform server-side rewriting (i.e., altered-dom) prior to serving the root memento. Alam et al. [7] implemented provided an alternate approach to mitigating the archived representation rewriting problem using a client-side rerouting scheme through the use of Service Workers.

This work focuses on the transformation of TimeMaps, not the mementos themselves. The rewriting problem in previous work is pertinent to replay of URI-Ms, whereas what we accomplish is more expressive metadata of the mementos prior to and to mitigate issues with dereferencing URI-Ms. A goal of this work is to further involve the client in the aggregation process. Interaction with the aggregators through these sort of mechanisms will be a first step in accomplishing this, as described further in this work.

In previous work [91], we highlighted an issue of URI-collision in the realm of

personal Web archives wherein (for example) both a login page and the authenticated content of a live Web application may reside at the same URI-R (Figure 10). We [99] extended this work by identifying personalized representations of mementos and providing a mechanism to navigate between additional dimensions beyond time. As personal Web archives proliferate and are at some point aggregated into multi-archive TimeMaps (cf. a TimeMap from and containing only listings from the archive itself), it would be useful to distinguish URI-Ms that represent personalized mementos, mementos that were originally behind authentication, and mementos in personal web archives that require additional considerations and mechanisms to access.

3.2 PRIVACY AND SECURITY

In preserving private Web content, issues of privacy and security arise when this content is stored and accessed via either replay or through the archival metadata representative of the captures themselves or even the archival holdings. For example, to expose that a capture exists for a URI-R in a private Web archive without necessarily exposing the capture's contents might encourage those trying to illegitimately access the private capture from proceeding, knowing their efforts might not be in vain. With one objective of this work being to facilitate aggregation of these captures, previous work dealing with privacy and security as it relates to Web archives needs to be evaluated to inform the design decisions of the framework.

When examining previous work performed in the realms of public, personal, and private Web archiving, it is useful to consider the issues of privacy and security of access. In Section 3.2.1 we discuss previous studies on the current practices performed by individuals' Web archives. In Section 3.2.2 we consider how access control practices are performed and can be adapted to personal and private Web archives from both the institutional and the live Web perspectives.

3.2.1 PRIVACY OF SOCIAL MEDIA CONTENT

Marshall and Shipman [121] surveyed Facebook users on Mechanical Turk with varying opinions on ownership of content that a user posts to Facebook. Over half of the users answered that public institutions should not archive Facebook, with one respondent stating that the content did not belong to Facebook or interested archiving institutions and another respondent stating that archiving institutions should

not proceed without user permission. Other users were vehemently against institutional Web archiving of Facebook content stating, "Whether it is is public or not, institutions really should not have a right to archive personal content." Users also said that to limit the archiving process only to public content changes the nature of the archives and "might ensure an anodyne source of historical information, less informative than a local newspaper."

Lindley et al. [116] interviewed Web users, particularly about their online habits in social media. Users expressed active efforts to separate their personal and professional personas, often using pseudonyms to accomplish this. For example, a user described her Pinterest persona as "housewifey" and not representative of her professional identity. Another user, an amateur photographer, stated of his pseudonym-associated Flickr account that the disassociation was a "public private thing" and that the Flickr persona, "isn't really me."

Kapil [90] highlighted methods where the OAuth 2.0 protocol may fail if not comprehensively implemented. He stated that the protocol is inherently insecure in that the security measures it specifies to use are optional and thus, often not comprehensively implemented. By iteratively focusing on each individual step of the authorization procedure (as illustrated in Figure 27), Kapil enumerates a method for each step to compromise the assurance of the protocol's security. For example, in attacking the connect request (the initial interaction between the entities in Figure 27), Kapil uses the Cross-Site Request Forgery (CSRF) approach. By creating a dummy account for the authorization provider/server and a dummy Web page to initiate the authorization, Kapil's procedure produces a token that is associated with the credentials entered with the provider. This approach can be mitigated on the provider's end by preventing CSRF for being used as the basis of logging in and out of an account and on the connect page (that initiates the transaction) by requiring requests to come from the user and not through a CSRF redirect. This can be accomplished by requiring a CSRF token in the first request, but the OAuth 2.0 specification does not state this requirement.

3.2.2 ACCESS CONTROL IN WEB ARCHIVE PRACTICE

Various Web archives have implemented a means of access control for their holdings. Two such examples are rudimentary password protection using a basic authentication mechanism and another of restriction of access based on location.

In early 2017 the UK Web Archive (UKWA) instituted a change in their OpenWayback instance, limiting what parts could be accessed over the Web. When accessing URI-Ms at this archive, e.g., `https://www.webarchive.org.uk/wayback/archive/*/http://www.example.org`, a user would receive a Web page stating that the memento can only be accessed from their "Legal Deposit Library reading room" (Figure 29). Using `curl` on this same URI-M returns an HTTP 451 (Figure 28), a status code indicative that the resource is unavailable for legal reasons [42]. Accessing this same capture while on-site at the archive permits access. In early 2018, the UKWA began migrating [172] to using an adapted version [167] of the Python-based pywb replay system to enforce these access restrictions in a more systematic manner.

```
$ curl -I
https://www.webarchive.org.uk/wayback/archive/*/http://www.example.org
HTTP/1.1 451 Unavailable For Legal Reasons
Date: Wed, 25 Oct 2017 04:39:35 GMT
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=utf-8
Transfer-Encoding: chunked
Set-Cookie: JSESSIONID=823BD09DF8DD489087763640A8150023; Path=;
HttpOnly
Content-Language: en
```

Fig. 28: Accessing a URI-M at UKWA using curl returns an HTTP 451 status code.

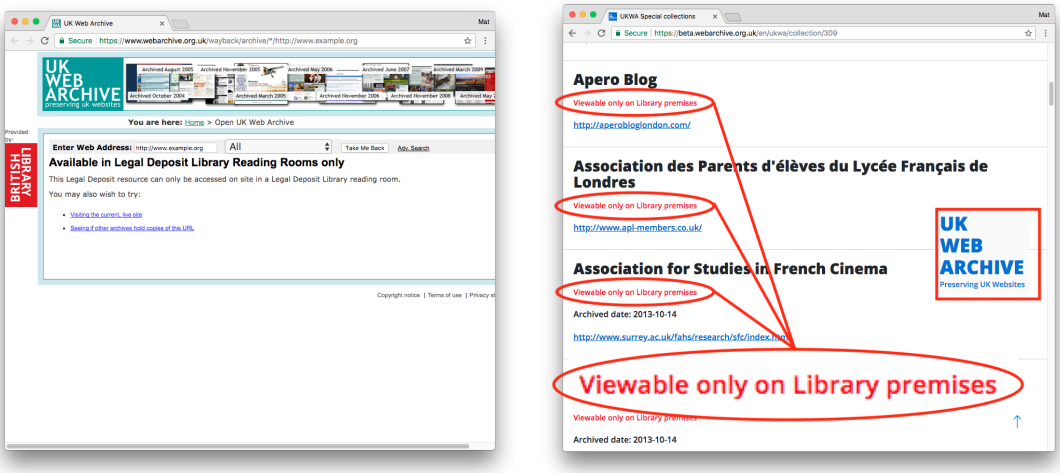


Fig. 29: Accessing a URI-M at UKWA using a browser returns an an interface informing the user that the URI-M can only be accessed on-site. The left screenshot corresponds to the HTTP 451 corresponding to Figure 28 when accessed using a Web browser whereas the right image corresponds to UKWA's recently collection-based replay interface displaying a message that access is limited to on-premises users.

3.3 COLLABORATION USING WEB ARCHIVES

Digital humanities scholars are interested in creating, curating, and sharing collections of Web archives but barriers currently exist that prevent members of a group of scholars from collaborating. For initiating crawls, for example, Dr. Liza Potts,

a Digital Humanities researcher at Michigan State University, wishes to use tagging as directives for automatic crawling (e.g, #crawlone, #crawlevery10minutes) and be able to sort archived pages by time or tag. Their use cases anticipate using familiar technologies to archive like writing a list of metadata for an archival target to a Google Doc with values like username, datetime, URI-R (live Web), URI-M (archive page), and associated tag. This Google Doc could then be used for sharing and sorting. To facilitate collaboration, these same scholars want to create small archived collections that can be shared among a small group of researchers.

Collaboration by individuals in Web archiving frequently involves centralizing to an institution. For instance, the recent collaboration on the Cobweb [161] project between the California Digital Libraries[1], Harvard University, and UCLA Library[2] involves a URI-R submission process to Archive-It, the latter who performs the preservation procedure. This preservation by-value procedure is common in calls for individuals to "archive the Web" through a URI nomination procedure. In our proposal we are working toward a more decentralized collaboration approach of collaboration by-value for an aggregate resilience of the collective picture of the Web instead of relying on a centralized institution to both perform the procedure but also to be solely responsible for its availability.

PANDAS is a system developed by the National Library of Australia that provided tagging to Web archives including restrictions by date (embargoes), authentication, and by IP address and is implemented via Apache's .htaccess file [136]. This system provides no fine-grain access control and suffers from other scale issues but was used as a basis for consideration in OpenWayback's implementation of access control[3]. Niu [129] examined the Australian PANDORA archive among ten other Web archives to compare the functionality and personalized-based features offered to users for personal web archiving. These features included comparing Web archive access methods such as lookup-by-URI as one method offered to users. Access in terms of means of lookup will be investigated in the context of private Web archives in our research, for which the URI colliding issue remains. iProxy provided users a means of archiving and replay with access parameters that extended URLs with commands for retrieval [141]. Because of the URI colliding issue, a similar extension

[1] https://www.cdlib.org/

[2] http://www.library.ucla.edu/

[3] https://web.archive.org/web/20090209140507/http://webteam.archive.org/confluence/display/wayback/Exclusions+API

of URIs will be needed for lookup in private Web archives whose content was behind authentication on the live Web.

3.4 ARCHIVING: PUBLIC VS. PRIVATE VS. PERSONAL

In this proposal we put describe a framework for aggregating public and private Web archives. To aggregate these different classes of archives, it helps to first understand what each classes comprises. As adding the "personal" aspect to a class of archives, the three classes are not necessarily mutually exclusive. In this Section we describe related research focus on each class.

3.4.1 PUBLIC WEB ARCHIVING

Brunelle [44, 50] proposed and evaluated a model for more comprehensive preservation of Web pages through identifying "deferred representations". The work used a large-scale public data Web archive collection as a gold standard basis in both replicating the original approach (through sampling) as well as the approach with supplementary URI-R discovery. This directly relates to RQ1 (Section 1.4) through a mechanism of URI surfacing. We can extend on this work to both improve on the deferral procedure through leveraging a browser medium for capture and also consider content beyond the original scope of the gold standard collection, like content behind authentication. The latter will introduce additional challenges, as content that requires parameters other than a URI for access is inherently more difficult to preserve.

Access to Web archives is a fundamental theme in accomplishing the work to be completed in this proposal. Ben-David and Huurdeman [25] described accessing Web archives through mechanisms beyond retrieval by URI. They juxtapose their searching techniques to the conventional access pattern of initially "vertically surfing" (accessing a URI at a point in time) then "horizontally surfing" by following links from the initial page. This paradigm correlates with the method of retrieving a single document from an analog archive. "Access to Web Archives has remained tied to the Web's early user engagement practices", they said, "of surfing and browsing and not searching", citing that most Web archives are not searchable. Their WebART project prototype (originally developed in Huurdeman et al. [78]) allows full-text search of the Dutch Web Archive and provides an aggregate view of the Dutch Web – a shift from access-by-URI to considering the whole Web archive as a unit of

analysis. While the preliminary work to be performed in this proposal is focused on access by URI, users will likely wish to access their private, personal, and even institutional public Web captures by alternate means. Potential means may stand to both be complementary and independent of accessing using Memento.

Ben-David and Huurdeman's interfaces for exploration of Web archives beyond URI are not unique to their prototype, as other archives are working on adapting to support search interfaces. For example, Costa and Silva [54] showed through juxtaposition to search engine usage that Web archive users prefer full-text search instead of search by URI. They also found that temporal navigation is not often used for restricting searches except for the preference toward the oldest documents.

AlNoamany et al. [14, 12, 13] used access logs they acquired from the Internet Archive to analyze what users and robots were accessing and through what method they were accessed. They found that 82% of the sessions they identified as humans accessing IA to be accessing the archive through referrals from other pages in the archive. They compared this to what they identified as robots accessing the archive where only 15% of the accesses they attributed had a referral from another archived page. By identifying humans as accessing certain content, the authors were able to infer what the archive's users thought were important enough to revisit. In this proposal we focus both on being able to re-access these preserved pages but also to aggregate and make accessible personal and private captures, which was out of the scope of the studies performed by AlNoamany et al.

3.4.2 PRIVATE WEB ARCHIVING

Brunelle et al. [45] discussed (as mentioned in Section 1.3) private Web archiving from a non-individual context. In this work, the authors described archival crawling scenarios by the MITRE Corporation to preserve the contents of their corporate Intranet. In some cases, a crawler preserved sensitive information, requiring the resultant WARC file to be deleted in lieu of a process to selectively remove particular captures from WARC files. In other instances, the shortcomings of the crawler not possessing the credentials to access privileged resources had a dramatic effect on the coverage of the crawl. In this same light, the lack of technical capability of Heritrix to execute and archive JavaScript-reliant representations prevented many pages from being comprehensively preserved. Brunelle et al.'s study proves relevant to the research in this proposal in that the ramifications of preservation of private

information has greater consequence beyond personally identifiable information being exposed, as is often cited as the need in individuals archiving the private parts of the Web.

Rauber et al. [142] discussed privacy issues in archiving private web content and provided a way to programmatically identify when web content contains information that requires special handling when archived. His discussion on the ethical implications of preserving this content and the current practice of access control exhibited by institutional web archives further justifies the need for a proactive means of access control instead of after-the-fact identification of private content content in web archives.

The Snowden Archive-in-a-Box project [115] is an autonomous version of the the Snowden Digital Surveillance Archive. The project uses a Raspberry Pi single-board computer along with other hardware and a data set containing files leaked onto the Internet by Edward Snowden to allow browsing of the files without a user fearing being surveilled. This use case highlights access as being the problematic factor beyond the base case of the content being sensitive.

Wang et al. [171] suggested a role-based access control system stemmed on proximity in a social networking context for automated inclusion for access. This approach is borrowed in our framework to regulate group access; e.g., when access is limited to those in a proximity, like within an IP address range, the authorization procedure need not be repeated but rather, an access token can be reused with a two-factor authentication-like scheme. This scheme can also be utilized to prevent access using this token beyond the IP address range.

Creators of Web content may consider parts of the sites they curate to contain private content despite being publicly accessible. While the crawler at Internet Archive may capture this content regardless, the Internet Archive has stated that it is not interested in offering access to web sites whose authors to not want their materials in the collection [80]. A site author may provide exclusions to archival content for their domain using robots.txt.

Marshall and Shipman [120] surveyed individuals using Mechanical Turk on their opinions of institutions preserving personal Web contents, namely, the Library of Congress Twitter set donated in 2010. The latter evoked a response where cultural importance was often deemed inversely related to the level of personalization of the content if archived by institutions. The authors also expressed concerns of the

respondents of losing access control of content they thought was important but still had aspects of personalization. This may be juxtaposed to earlier work by Marshall and Shipman [119] on content ownership of a photo posted online where a non-consenting individual from an adjacent party where the photo was taken was clearly visible and identifiable. In the hypothetical scenario where this photo is preserved and the individual who posted the photo retains access control, the background individual who the preserved Web content is partially "about" has less of a grounds of ownership and thus obtaining any access control to the preserved content. These two works relate to the scenario in Chapter 1 of our need to preserve personal photo-based content where we are the bearer and thus accountable for the level of publicness if shared.

3.4.3 PERSONAL WEB ARCHIVING

Abrams et al. [3] described a bookmarking system he labeled as personal/private "archiving" but which was more of a preservation-by-reference approach where contemporary archiving is preservation-by-value, in addition to maintaining a reference key for lookup and replay. He reiterated this point with the admittance that "bookmarks aren't great describers of the actual content [of the Web page]" reinforcing the link rot that occurs when a representation for a URI has changed.

Thelwall and Vaughan [164] explored the bias of the collection of web sites preserved by Internet Archive as a selection of the "whole Web". This evaluation did not extend to the private live Web for which an even larger bias exists, as the overwhelming majority of content preserved by IA is from the public live Web. Gomes et al. [71] evaluated biases in web archive corpora that occur when the process of choosing which sites to archive in focused crawls is automated with a criteria basis. His consideration of the user in developing access models is relevant in the user-based access models that we are developing for aggregating private and public web archives.

Marshall [117, 118] enumerated examples of personal digital archiving extending beyond Web archiving. The usage patterns give real-world scenarios of how individuals preserve and access their digital content including the distribution of collections, what sort of content is preserved, and the role of the storage medium in ensuring future access. With the audience of this framework ultimately being these same amateur archivists, Marshall's patterns help to understand the technical needs of the users in developing the framework.

In our previous work [105, 99, 100, 48] we highlighted and evaluated the digital preservation capabilities of tools used to preserve content on the live web, particularly in respect to JavaScript. These works accounted for archiving content on the public live web though much of the private live web is dynamic and JavaScript-driven, proving the likelihood of a higher degree of damage in mementos [46]. We have preliminarily used browser-based tools [107] for a subset of the web archives we created from the private live web to generate private web archives.

Strodl et al. [162] described a user-driven framework for digital preservation that facilitates individuals' preservation of private digital content using best practices. Their software prototype predates and shares similarities with our prototype [102] to encourage users to archive their private web content by removing technical barriers in the preservation software. Strodl's work abstracts the access issues that will need to be addressed when the implementation of the framework creates data akin to the sort he describes.

3.5 SUMMARY

In this chapter we provided a review of recent related work that is relevant to the research being performed in this candidacy proposal. In Section 3.1 we discussed recent work relating to Memento with a focus on aggregation and dynamics that others have explored beyond the original specification. Section 3.2 provided an overview of recent investigations of privacy and security as applicable to Web archives. Section 3.3 described the rudimentary approaches currently used for collaboration using Web archives, with which we extend on in an accessible way in this proposal. Section 3.4 outlined a means of distinguishing personal, private, and public Web archives and the various gray areas where each may exhibit traits of multiple classes and how that makes aggregation non-trivial.

CHAPTER 4

PRELIMINARY RESEARCH

In this chapter we will outline our research thus far [92, 106] towards a framework to aggregate public and private Web archives. The chapter is organized through the various processes anticipated within a Web archiving workflow that this research addresses. The vastness and ever-increasing amount of content on the live Web limits what is preserved for posterity. Institutional Web archives perform the bulk of the Web archiving but individuals often create content that is unique, niche, and personally important to the user and may be missed by these institutions' efforts. This chapter addresses preliminary work related to Research Questions 1-3:

> **RQ1:** What sort of content is difficult to capture and replay for preservation from the perspective of a Web browser?
>
> **RQ2:** How do Web browser APIs compare in potential functionality to the capabilities of archival crawlers?
>
> **RQ3:** What issues exist for capturing and replaying content behind authentication?

In Section 4.1 we describe our initial efforts in enabling individuals to perform Web archiving for content they feel is important on the live Web. Oftentimes content is not preserved, even by institutions, because of difficulties in capturing the resource representations (RQ2, RQ3). In Section 4.2 we describe studies we have performed in evaluating the archivability of certain sorts of resources on the Web (RQ1), assigning metrics to both evaluate resource importance as well as creating benchmarks to determine where the difficulties lie for Web archiving tools to preserve live Web contents.

By encouraging individuals to preserve the Web, it follows that more individuals' archives will be created. However, individuals' archives are more likely to disappear than an institution's. Thus, encouraging individuals to share their captures facilitates the creation for a more comprehensive picture of the Web being available when

accessed using personal Web archives in the future. In Section 4.3 we describe research we have performed and tools we have created to facilitate individuals being involved in creating this more comprehensive picture using their captures.

4.1 ENABLING THE PERSONAL WEB ARCHIVIST

As the Web has evolved and society has deemed it culturally significant and thus should be saved, it has drastically changed form from the Web of the past. What were once static pages are now dynamic with content often hidden and only requested and displayed based on a user action [44]. Facebook.com, for example, only shows temporal details on-demand, obscuring the potentially vast amount of content until it is explicitly requested (Figure 2). The content being displayed or at least referenced in the DOM is often a prerequisite for it being comprehensively preserved.

Other Web pages may be inaccessible or inappropriate for institutional archives and their tools to capture. For instance, my born-digital baby photos in Figure 1 ought to not be the responsibility of the institutional archives to preserve despite being on the live Web. However, as a Web user, I feel this content is extremely important and thus, despite not being the bearer of these photos (Google is in this case, per Chapter 1), it ought to be my responsibility to preserve them. Figure 30 describes this issue of scoping the appropriateness of various kinds of Web archiving. Here, a user may want to preserve their Facebook captures and Private Bank Record captures in separate but aggregate-able (blue box) private Web archives and, despite the personal natures of their captures of a site like `cnn.com`, are willing to allow aggregation of these particular captures (green box).

The issues of the Web being dynamic beyond the capability of the institutions' tools and content being sensitive and in need of an individuals' efforts to ensure its posterity lead us into further investigations in enabling individuals to preserve the Web using capable tools with privacy considerations in-mind.

4.1.1 ARCHIVING SOCIAL MEDIA

This section addresses Research Questions 2 and 3. We initially performed an investigation into personal Web archiving, leveraging tools with which Web users were already familiar. Contemporary Web users conventionally view the Web using the means of a Web browser. Our early work targeted the preservation of social media Web sites (particularly, archiving Facebook [108] to create a single private archive per

Figure 30) using browser extensions to leverage the browser interface with which users would already be familiar. Tools tailored to preserve a particular site often break when the target site changes (e.g., a tool for scraping Facebook begins to fail when they change their HTML [124]), so we created an extensible framework for preserving content behind authentication [91], with a focus on social media sites. The primary method for preserving the content, however, was inconsistent with standard practice and formats, i.e., we stored the resource representations (HTML, CSS, images, etc) into individual files on the local file system.

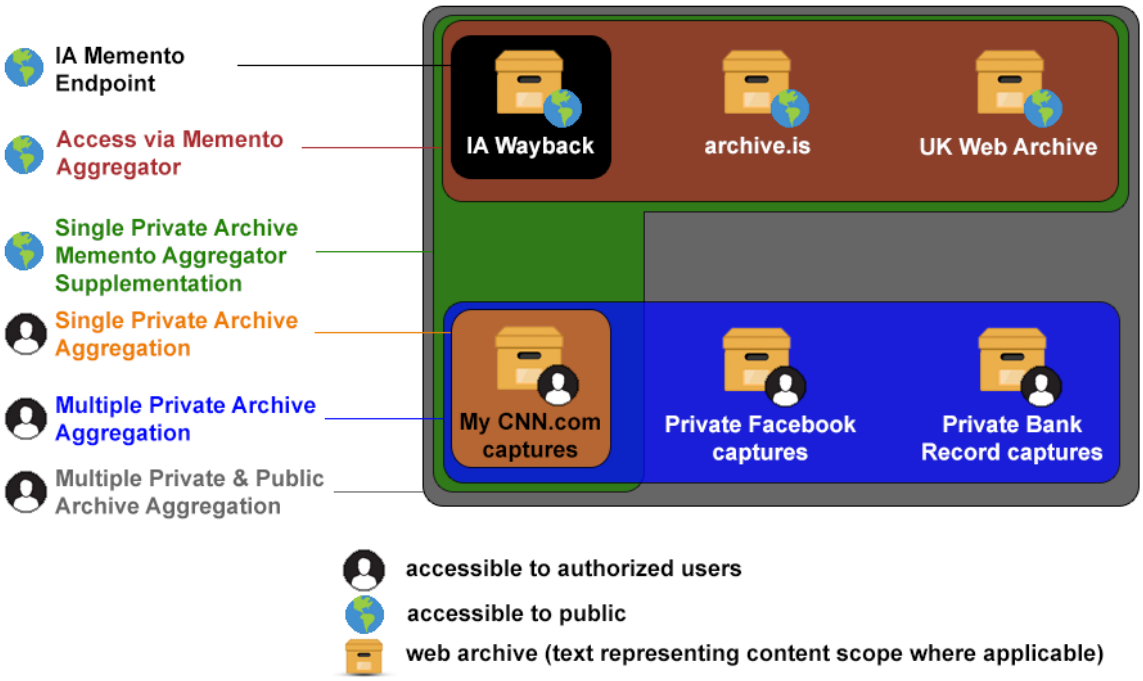


Fig. 30: Various currently existing archives in the Web archiving spectrum are limited to the part of the Web they *can* or appropriately *should* preserve. An individual archive (black) may be aggregated with other public Web archives (maroon) but Memento aggregators do not typically include personal captures of the public Web (green is not performed in-practice), despite the aggregation potentially facilitating a more temporally complete picture of the Web.

4.1.2 WARCREATE

We discovered that making the browser a part of the preservation process facilitated users preserving the part of the Web they cared about. In 2012 we developed WARCreate [107, 109], a browser extension for the Google Chrome Web browser.

The purpose of the extension is to make the standard format for preserving Web pages, the WARC format (Section 2.4.3) [82], more accessible for preservation via generation of the capture from a Web browser. Unlike most methods for generating WARCs, WARCreate mitigates the technical overhead to accomplish this by allowing the user to preserve Web pages to WARC files without leaving the browser. Prior to WARCreate being developed, the bulk of users' direct efforts in creating WARC files (cf. indirect efforts like submitting URIs) was through running Heritrix crawls (Section 2.4.2). Delegation of the archival process by passing the target to be archived by reference (i.e., supplying a URI-R, see Section 3.3) introduces the potential for a difference in content of what a user sees in their browser and what is captured by the archiving tool (RQ1). This potential for a representation to be different when passed by reference formed the basis for our further research in facilitating the capture of the live Web by extending tools a user already uses in their daily workflow to allow them to "Archive What I See Now" (Section 3.3) [173]. Additionally, WARCreate's privileged access as a browser extension to what a user sees in their viewport, even content behind authentication (RQ3), allows it to capture content inaccessible to Internet Archive and Heritrix.

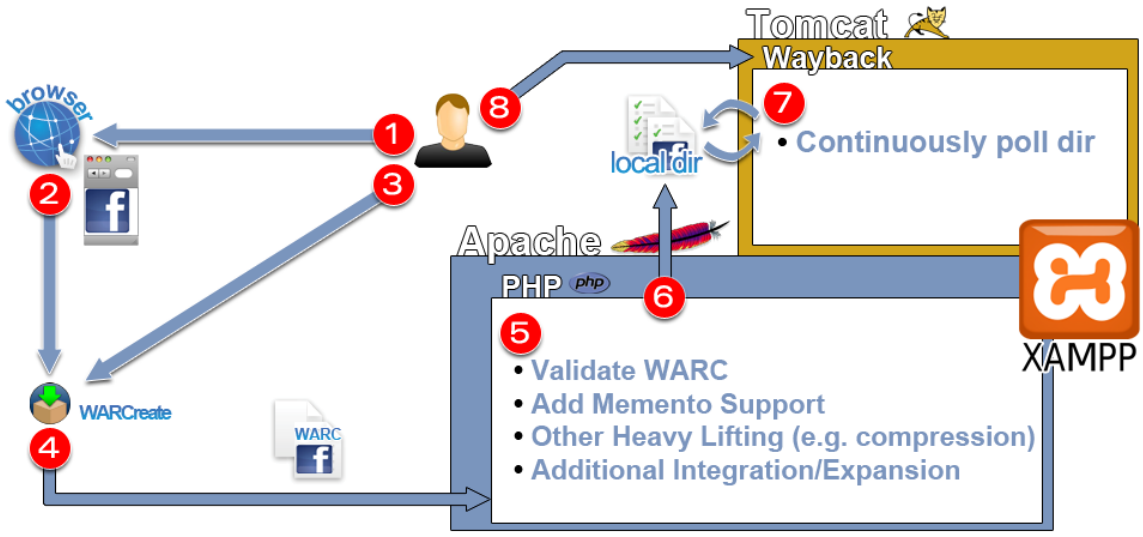
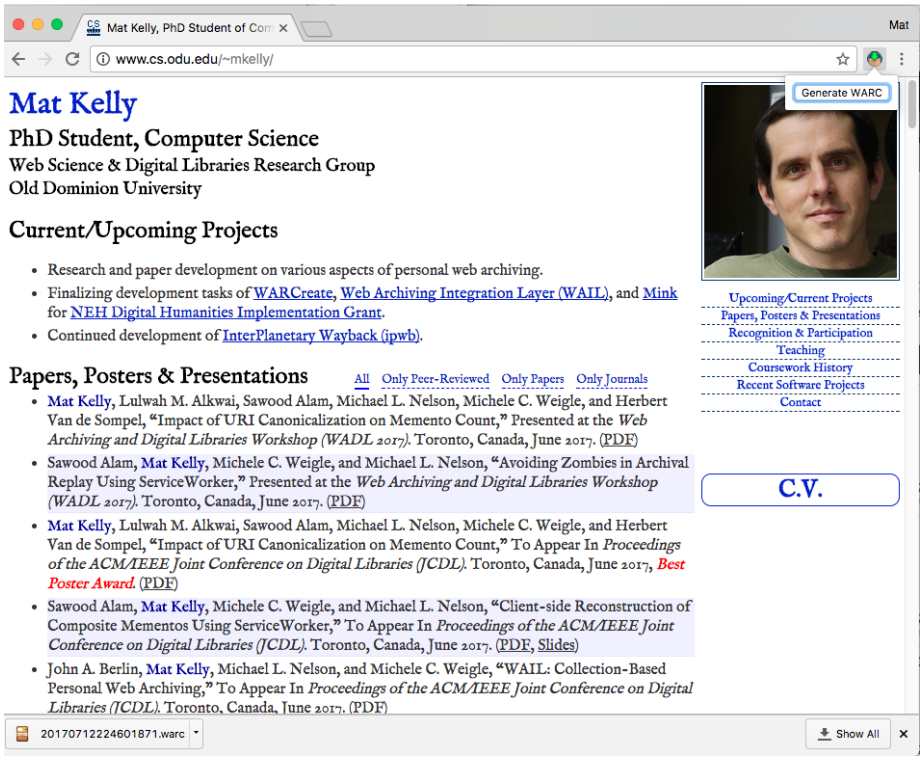


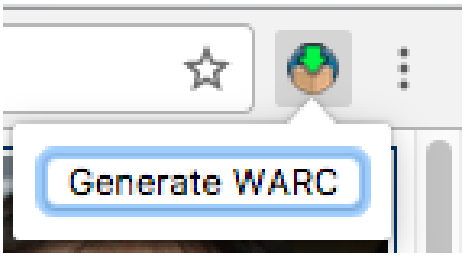
Fig. 31: When developing WARCreate [107], a local server instance was originally required to write to the file system. When browsers became more capable, the server components were repackaged along with the additional inclusion of Heritrix and deployed as Web Archiving Integration Layer (WAIL) [102, 103].

4.1.3 WEB ARCHIVING INTEGRATION LAYER (WAIL)

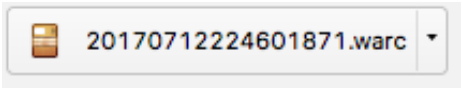
Creating a tool to capture what a user sees in their browser was not straightforward in 2012. For the sake of security, browser extension APIs allow limited access to both what is being read through the network and interpreted by the browser as well as the local file system [94] (RQ2). The File API [140], still in the draft stage and not yet fully supported by any browser [56], also provided no reprieve, as files produced by browser extensions were sandboxed and inaccessible from the rest of the file system. To counter this, rather than rely on a central Web-based endpoint, we leveraged the cross-platform and desktop-based XAMPP [58] to act as a local bridge to allow the user to write WARCs to their local file system (Figure 31). In doing so, we also opened the opportunity to allow users to use desktop-based applications relating to Web archiving. We adapted configurations for OpenWayback and Heritrix and rewrote a tailored Graphical User Interface (GUI) to create Web Archiving Integration Layer (WAIL) [102, 103] to encourage users to create personal Web archives with the ease of entering a URI and selecting an "Archive Now!" button (Figure 33). This opened the door for users to create their own personal Web archives without the normally required technical overhead.



(a) WARCreate Browser with highlighted details in sibling subfigures.

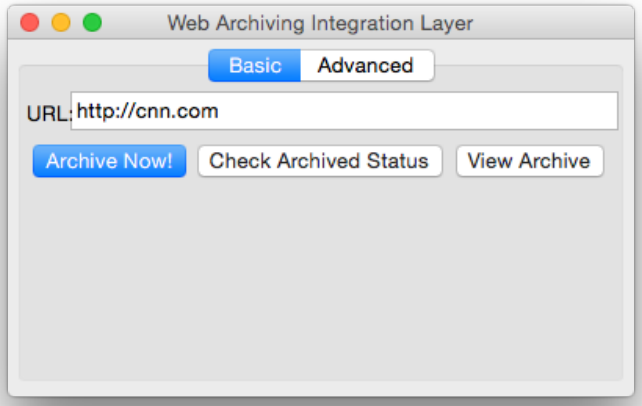


(b) The WARCreate popup consists of a single button interface for simplicity to encourage preservation without complication.

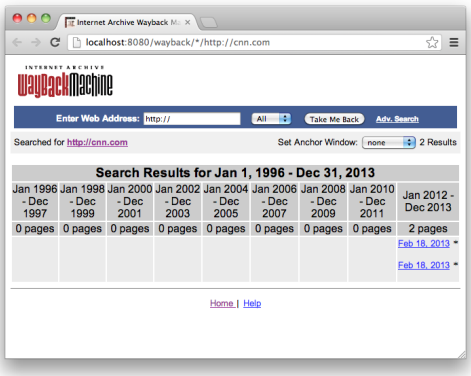


(c) Native Chrome UI provides direct access WARCreate-generated WARC file.

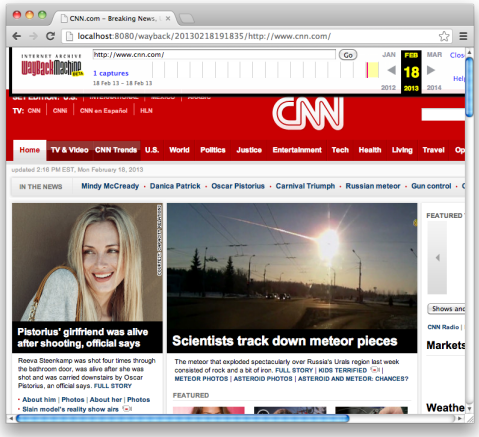
Fig. 32: WARCreate is activated by a user clicking a button bar icon when on a page for which they want to create a WARC. The figure shows the placement and context of the icon with the single button (a) to generate the WARC after clicking the button bar icon (details in (b)) and the native Chrome downloaded file interface providing immediate access to the downloaded file (c).



(a) The original WAIL Interface (ca. 2012)



(b) Capture listing access through WAIL's included OpenWayback



(c) Viewing CNN capture in local OpenWayback

Fig. 33: Web Archiving Integration Layer (WAIL) allows users one-click access to preserving live Web URIs. This figure shows a user entering a URI in the native (macOS) desktop application interface (a), viewing the capture listing in the bundled OpenWayback interface once the capture procedure is complete (b), and viewing the memento being served from the OpenWayback instance (c) included in their local WAIL.

Archiving content from the browser provided a unique perspective to the Web archiving process. Content that is otherwise inaccessible with by-reference delegation (i.e., instructing another tool to archive what is at a URI) could now be preserved.

On the other hand, those pages may contain sensitive, private, or personally identifiable information. As described in the scenario in Section 1.3, violating expectations of sensitivity may have side-effects and ramifications that affect other preserved content. On the level of personal Web archiving, a user has no easy way of knowing that content in a WARC is sensitive, private, and/or contains personally identifiable information. With the desire to facilitate preservation and tools to enable users to preserve content on the live Web that would otherwise go unpreserved, one goal of this research is to provide a means of allowing a user to specify these additional dimensions for their personal collections.

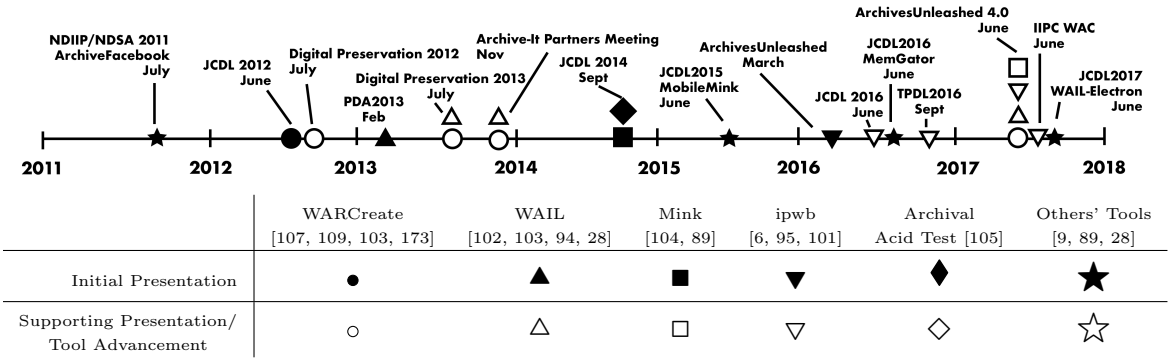
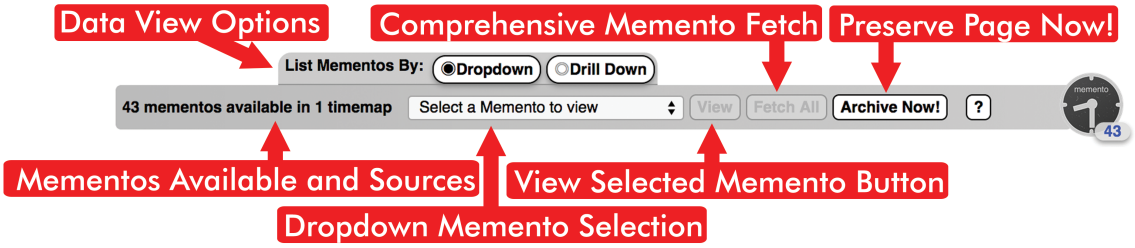


Fig. 34: In the course of this preliminary research, we have developed and extended numerous tools. Unlike most research software, these tools were publicly released and continually maintained. Further, these tools continue to inform further research in the area and provide the basis for extension, as applicable, to exhibit the roles of the mementities.

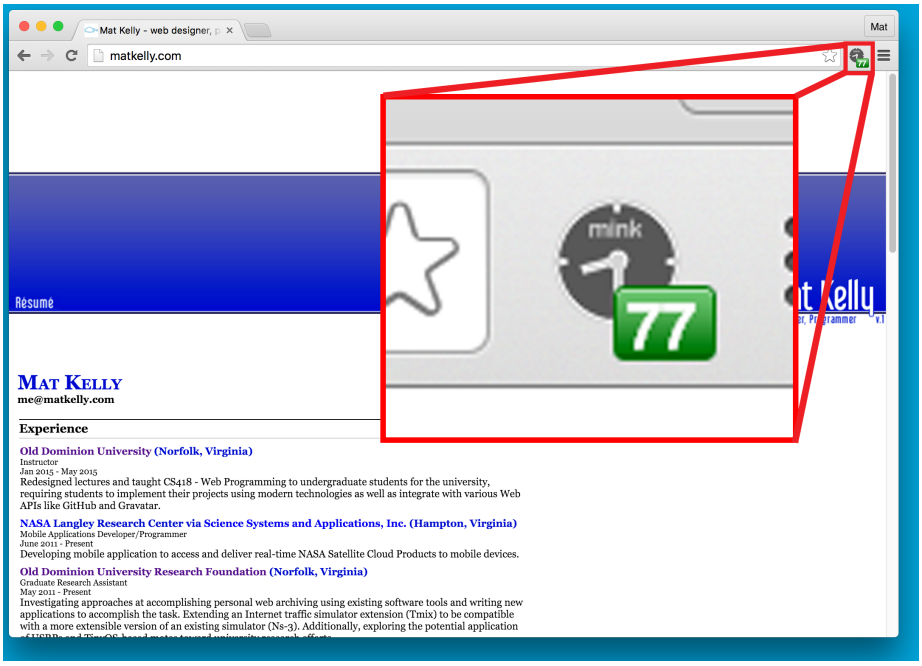
4.1.4 MINK

Building upon our work in enabling Web users to preserve content, we inverted our perspective on tool building to the realm of access of Web archives. The temporal gaps in only using a single Web archive as a source for the historical record (e.g., looking solely to Internet Archive for the Web's history) may be mitigated by including additional Web archives in the temporal picture. Previous work building on the Memento framework through Memento aggregation (Section 3.1) still left a large gap in bridging the live Web and the archived Web. We created an additional Web browser extension (informed by our previous creation of WARCreate) we named Mink [104] (an homage to "Minkowski space", which deals with three spatial dimensions and a dimension of time) in an effort to bridge this gap. As a

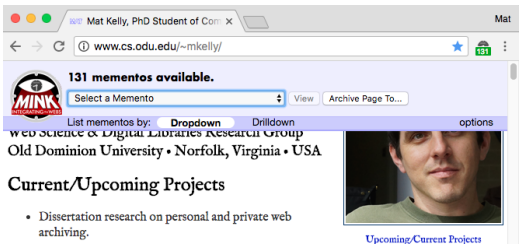
user browses the live Web, an indicator is persistently displayed in the user's browser (originally within the viewport (Figure 35a) but refined to be less obtrusive in the browser button bar (Figure 35b)) to indicate the quantifiable extent (i.e., number of mementos) to which the URI-R they are viewing on the live Web is archived. This is accomplished by the extension querying a Memento aggregator and reporting the memento count (which we later showed as a different and variable means of counting mementos [98, 97]). Selecting the Mink icon displays an interface in the viewport where a user may browse to any memento listed for the URI-R (using a dropdown or drilldown interface in Figures 35c and 35c, respectively) or submit the URI-R to multiple supported archives with a single click. Upon submitting a URI-R to an archive, the interface provides one-click access to viewing the memento. When viewing a memento created by either navigating through the list of available mementos or viewing the newly created memento, a button in the Mink interface allows the user to return to the live Web. This association is accomplished using the "original" relation type within the Link HTTP response header of the memento. In a continuation of this work [89], we later adapted the navigation-based archival querying and archival submission logic to an Android application named "Mobile Mink", which overloaded the native sharing function of the mobile operating system to integrate the live and archived mobile Web.



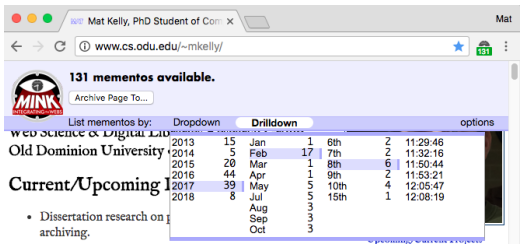
(a)



(b)



(c)



(d)

Fig. 35: The Mink browser extension displays the number of captures for a URI-R while you browse. The original interface included the indicator and interactive interface within the viewport (a) but was later moved to the browser's button bar (b) to be more persistent and less obtrusive. After the TimeMap for the URI-R has been acquired, the mementos can be accessed in the Mink interface through a dropdown menu (c) or a Miller column-style temporal drilldown interface (d).

The original implementation of Mink communicated with the Memento aggregator at `mementoweb.org`, but much like the issues of a changing API experienced with Archive Facebook, the API at the aggregator changed over time, causing Mink to break in its TimeMap parsing algorithm. We deployed an instance of Alam and Nelson's MemGator [9], as described in Section 3.1, at ODU in order to have a more consistent API as well in anticipation of customizing the set of archives requested, as explored in this proposal.

To relate back to *personal* Web archiving and to make Mink more useful for individual archivists, we later expanded on Mink [173] to allow for users to specify a custom source for aggregation (inclusive of their own MemGator deployment) and provide additional sources for Mink to use to perform its own after-the-fact aggregation, e.g., captures in a user's local WAIL installation would be aggregated with captures from the remote MemGator instance, their own local MemGator instance, and any other sources. Inclusion of a user's local capture aggregated inline with institution's captures provides a user with a better picture of how a URI-R has changed over time. Aggregation with personal captures in this manner, however, begets RQ5.

4.2 MEASURING ARCHIVABILITY

In previous work we measured the ability for Web pages to be archived. Being able to evaluate *archivability* from what was preserved and what is currently preservable with state-of-the-art archiving tools gives a basis for further challenges to be experienced in Web archiving by both institutions and individuals. In this section we describe five separate investigations:

- An evaluation of the change in archivability over time [100] (Section 4.2.1)
- An investigation to evaluate the impact of JavaScript on archivability [48] (Section 4.2.2)
- An "archival acid test" to determine the state of the art of institutional preservation systems (Section 4.2.3)
- A "memento damage" metric to evaluate the quality and potentially quality of an archive and a live Web page (respectively) [46, 47] (Section 4.2.4)
- An analysis of archives to determine personalization of the content within the capture [99] (Section 4.2.5)

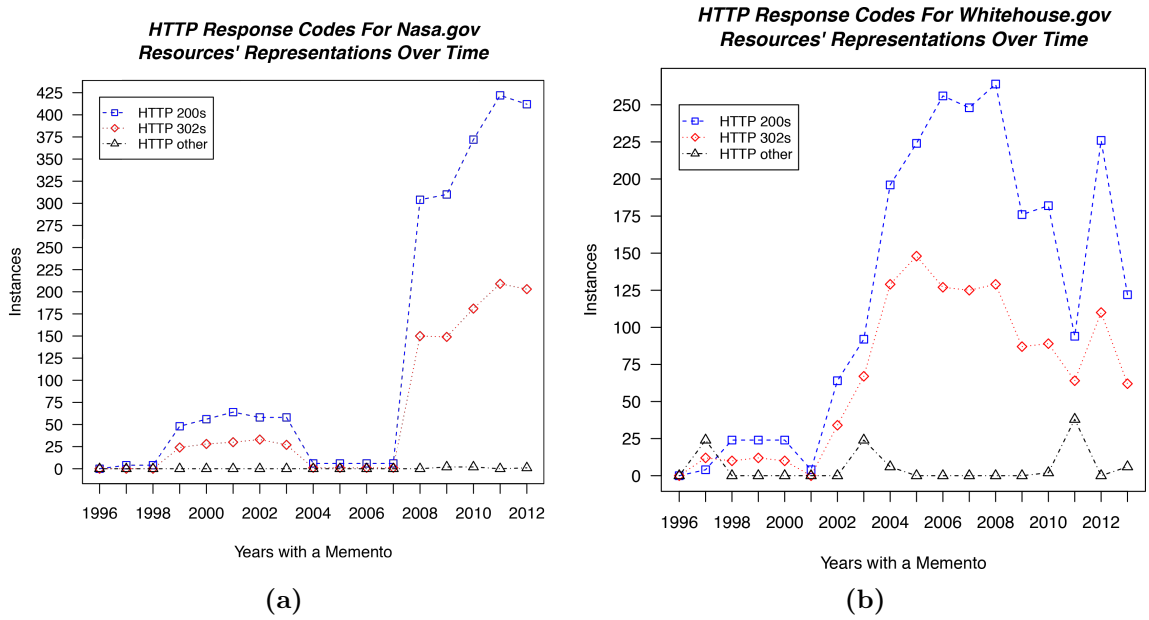


Fig. 36: The total number of URI-Ms to reconstruct a single memento for a year can be determined as the sum of each point for a chosen year. The Web page of `nasa.gov` (a) has a noticeably absent lull from 2004-2007 that corresponds to Figure 6 (with a single year temporal shift due to the sampling method). The preservation of the White House Web page (b) exhibits a different problem yet is briefly similar in that the count drastically changed. The sudden change in 2011 is the result of a set of CSS files not reaching the crawler horizon, which may have had implications on subsequent resource representations (embedded within the CSS) from being preserved.

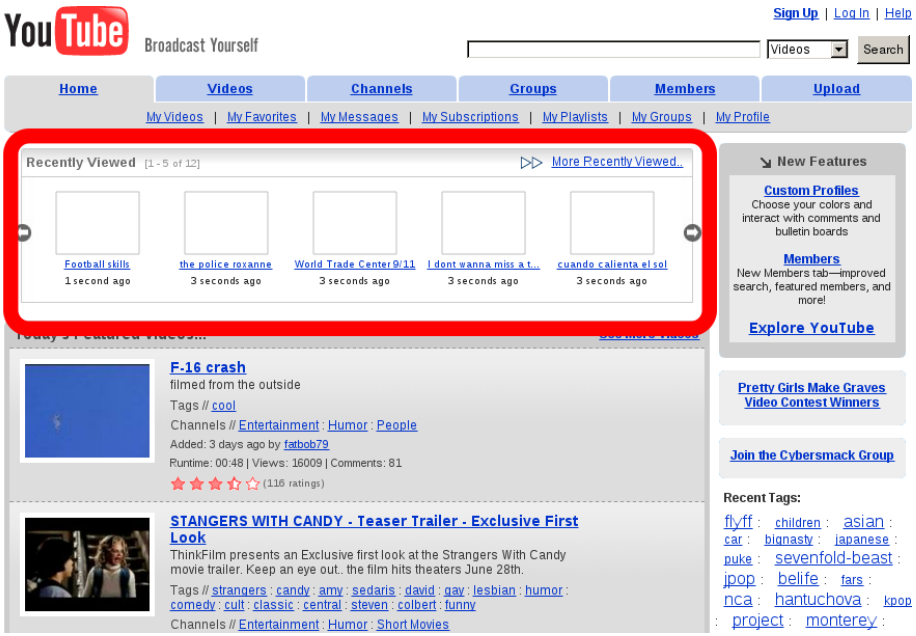
4.2.1 CHANGE IN ARCHIVABILITY OVER TIME

Even among the institutional grade Web archiving tools like Heritrix, we found that captures are not always complete due to missing embedded resources. We measured how the Web has changed in terms of archivability over time [100] by acquiring TimeMaps for the top 10 Alexa sites at the time of the study (2012). We found that some had a robots.txt file, which prevented Internet Archive from showing captures in their replay system at `archive.org` (Table III). The longevity of a URI-R was useful in evaluating how the changes in Web technologies have affected each URI-R's archivability. In particular, JavaScript's impact on archivability has been profound. Figure 37 shows a capture of `youtube.com` from 2006 with a subtle distinction in the display caused by missing resources that are missed due to the

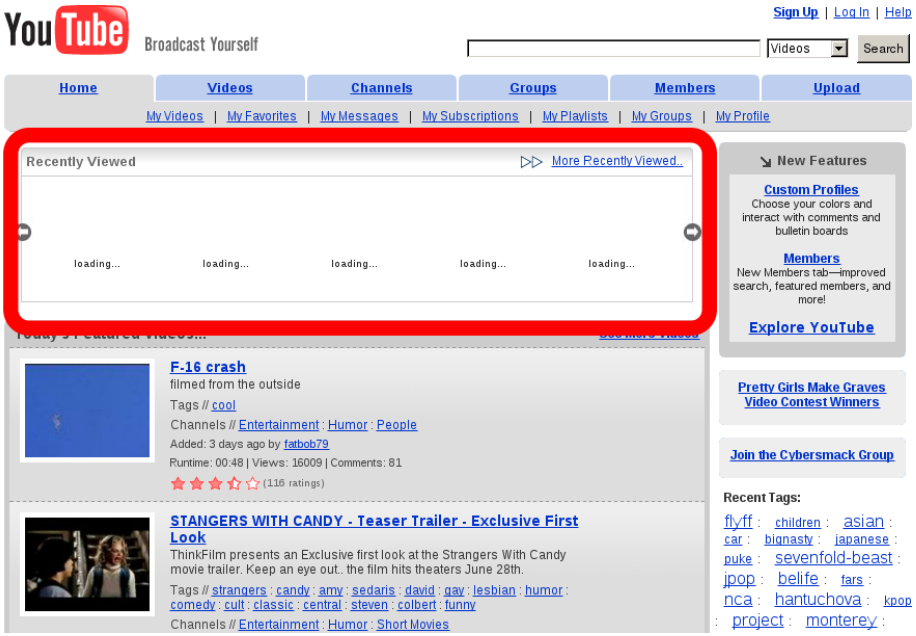
crawler's capability to process JavaScript at the time. Figure 38 shows the same URI-R but from a capture in 2011 (Figure 38a) and the causal chain of failure (i.e., one resource missing caused additional missing representations) that resulted from the memento attempting to fetch resource representations that were not preserved due to the capability limitations of the crawler (Figure 38b). As a sample of the effects of JavaScript over time, we plotted the number of resources for the URI-Rs `nasa.gov` and `whitehouse.gov`, two sites that are mandated to observe Section 508 [168] accessibility compliance for Web sites (and other Web accessibility initiatives [175, 53]) to indicate the trend of the number of missing resources for each URI-R over time (Figure 36). The dip in the plot of `nasa.gov` correlates with the annual screenshots in Figure 6 with a slight off-by-one shift due to the difference in annual sampling between the two studies. From this we concluded that the the archivability of a URI-R at a point in time is directly correlated with the number of resources; that is, the smaller number of resources between 2004 and 2007 was indicative of the un-archivability of the site between that time range, as evidenced by the completely black screenshots of the URI-Ms in that time range per Figure 6. This highlights a key difference in what browsers of the time saw compared to what the archival crawler at Internet Archive experienced due to a difference in capability (RQ1 and RQ2).

4.2.2 IMPACT OF JAVASCRIPT ON ARCHIVABILITY

With the recognition that archivability has changed as Web technologies evolved, we continued our investigation with a more focused investigation of the impact that JavaScript has on the archivability of Web pages [48]. Executing JavaScript on the client can potentially cause the representation to change with or without subsequent requests to a server for additional resources. We defined *deferred representations* as representation of resources that are difficult to archive because of their use of JavaScript and other client-side technologies. "Deferred" in this case refers to the final representation that is not fully realized and constructed until after the client-side representation is rendered. Because most Web crawlers do not have the ability to execute embedded JavaScript or other client-side technologies, the resulting mementos may only be partially operational or incomplete. For example, Figure 39 shows `http://maps.google.com` as it exists on the live Web, as archived in December 2012, and as archived in April 2012. The map in the middle is draggable,

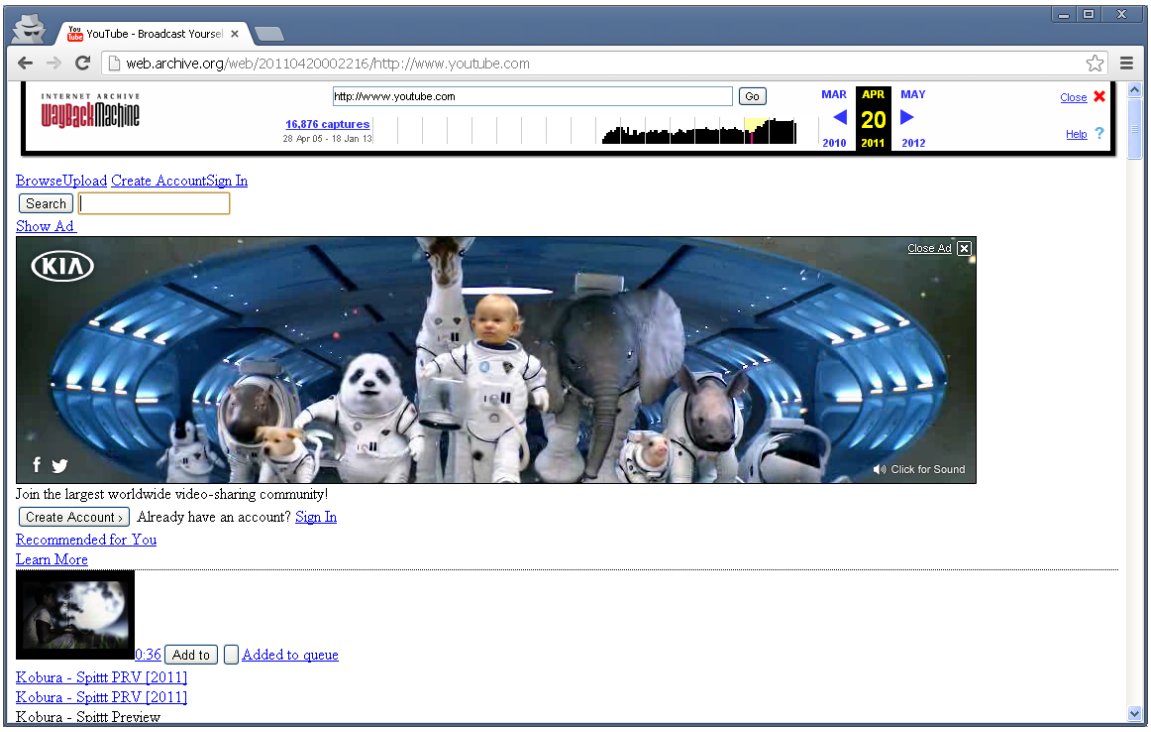


(a) Replay of YouTube with JavaScript enabled



(b) Replay of YouTube with JavaScript disabled

Fig. 37: A YouTube memento from 2006 shows a subtle distinction (circled in red) in display when JavaScript is enabled (a) and disabled (b) at the time of capture. The AJAX spinner (above each "loading" message (b)) is never replaced with content, which would be done were JavaScript enabled on capture. When it was enabled, the script that gathers the resources to display (blank squares in the same section of the site in (a)) is unable to fetch the resources it needs in the context of the archive. The URIs of each of these resources (the image source) is present as an attribute of the DOM element but because it is generated postload, the crawler never fetches the resource for preservation.



(a)

```
GET http://web.archive.org/web/20121208145112cs_/http://s.ytimg.com/yt/cssbin/www-core-vfl_OJqFG.css 404 (Not Found)
↪ www.youtube.com:15
GET http://web.archive.org/web/20121208145115js_/http://s.ytimg.com/yt/jsbin/www-core-vfl8PDcRe.js 404 (Not Found)
↪ www.youtube.com:45
Uncaught TypeError: Object #<Object> has no method 'setConfig' www.youtube.com:56
Uncaught TypeError: Cannot read property 'home' of undefined www.youtube.com:76
Uncaught TypeError: Cannot read property 'ajax' of undefined www.youtube.com:86
Uncaught TypeError: Object #<Object> has no method 'setConfig' www.youtube.com:101
Uncaught ReferenceError: _gel is not defined www.youtube.com:1784
Uncaught TypeError: Object #<Object> has no method 'setConfig' www.youtube.com:1929
Uncaught TypeError: Cannot read property 'home' of undefined www.youtube.com:524
GET http://web.archive.org/web/20130101024721im_/http://i2.ytimg.com/vi/1f7neSzDqvc/default.jpg 404 (Not Found)
```

(b)

Fig. 38: The 2011 capture of this YouTube.com memento (a) demonstrates the causal chain (Section 4.2) that occurs when a resource is not captured. The browser console at the time of replay (b) shows that a JavaScript representation that was embedded on the live Web page but was not preserved is used by subsequent scripts. Additionally, a missing CSS file (first line of (b)) prevents the memento from being styled as it was on the live Web. Other missing representations, like an image as detailed on the last line of (b), exacerbate the display issue.

Alexa Rank	Web Site Name	Available Mementos
1	Facebook.com	no mementos, robots.txt exclusion
2	Google.com	15 mementos 1998 to 2012
3	YouTube.com	7 mementos 2006 to 2012
4	Yahoo.com	16 mementos 1997 to 2012
5	Baidu.com	no mementos, robots.txt exclusion
6	Wikipedia.org	12 mementos 2001 to 2012
7	Live.com	15 mementos 1999 to 2012
8	Amazon.com	14 mementos 1999 to 2012
9	QQ.com	15 mementos 1998 to 2012
10	Twitter.com	no mementos, robots.txt exclusion

TABLE III: Alexa's 2012 Top 10 Web sites and available mementos obtained in January 2013 when evaluating the change in archivability of the Web over time [100].

allowing the user to plan. The April 2012 version of the page is missing UI elements and functionality (circled) and the interaction (e.g., panning and zooming) does not function. This is due to resources that would be loaded when the user clicks, but that are not preserved by the crawler. The December 2012 capture gives the facade of functionality when, in fact, resources on the live Web are being loaded.

Fig. 39: Google Maps as it exists on the live Web (a) and as a memento. The figure shows a deceptive representation with some interface elements being pulled from the live Web (b) while the annotated version of (b) shown in (c) makes it more evident that these resources are missing as compared to the live Web version (a).

We evaluated the impact of JavaScript on archivability by using a Twitter dataset consisting of bit.ly[1] URIs shared over Twitter (901 after filtering) and a dataset sampled from Archive-It (where the URIs are curated by humans). This summed to 960 URIs after filtering then sampling for relative similarly sized data sets between the two collections. From these data sets we discarded non-HTML representations, as they do not contain embedded resources when replayed. We then evaluated the

[1] `https://bit.ly`

complexity of the URIs in each of these collections by first considering the client-side (values following a # in a URI) and server-side (values following a ? in a URI) parameters as a value F per Equation 1. The URI complexity UC then can be determined with consideration of the URI depth (number of levels down from the TLD) and F (Equation 2). Using these equations we found $\overline{UC}_{Twitter} = 1.76$ and $UC_{Twitter_\sigma} = 0.312$ meaning there are nearly 2 URI parameters in the Twitter data set for each URI. For the Archive-It dataset, we found $\overline{UC}_{Archive-It} = 0.16$ and $UC_{Archive-It_\sigma} = 0.174$ meaning the URIs are mostly without parameters. Only 3 URIs from the Twitter data set had both self-side parameters and client-side fragments (i.e., client-side "parameters"). The Archive-It collection is a lower $\overline{UC}$ than the Twitter collection (Figure 40), supporting the theory that the human-curated Archive-It collection deals more with higher-level URIs than the shared links of Twitter.

$$F = max(|\text{client-side parameters}| \ , \ |\text{server-side parameters}|) \tag{1}$$

$$UC = \frac{|Depth| + F}{2} \tag{2}$$

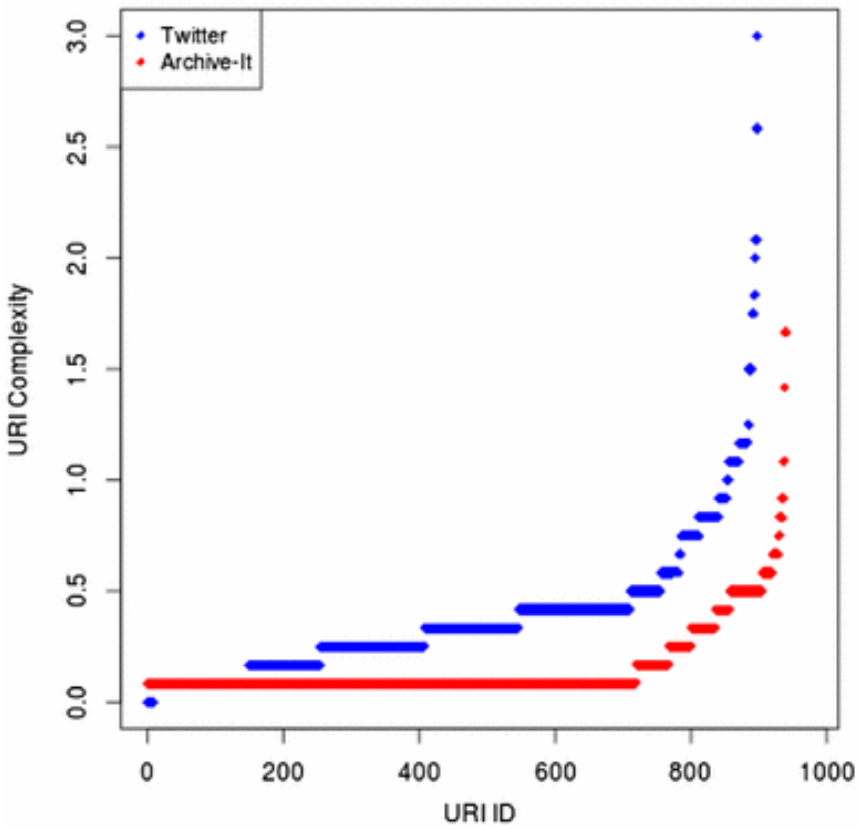


Fig. 40: URI complexity measure (UC)

To actually measure the impact on the mementos and embedded resources beyond the URIs in the collection, we established a content complexity measure CC, simplified for consideration of JavaScript (Equation 3). The Twitter set had a $\overline{CC}$ = 4.78 with CC_σ = 16.23 and the Archive-It set, an average $\overline{CC}$ = 2.16 with CC_σ = 6.87. The Archive-It set had, on average, approximately half as many `<script>` tags as the Twitter set and a CC_σ that is half of the Twitter set.

$$CC = \Sigma \text{ script tags } \epsilon \text{ HTML} \tag{3}$$

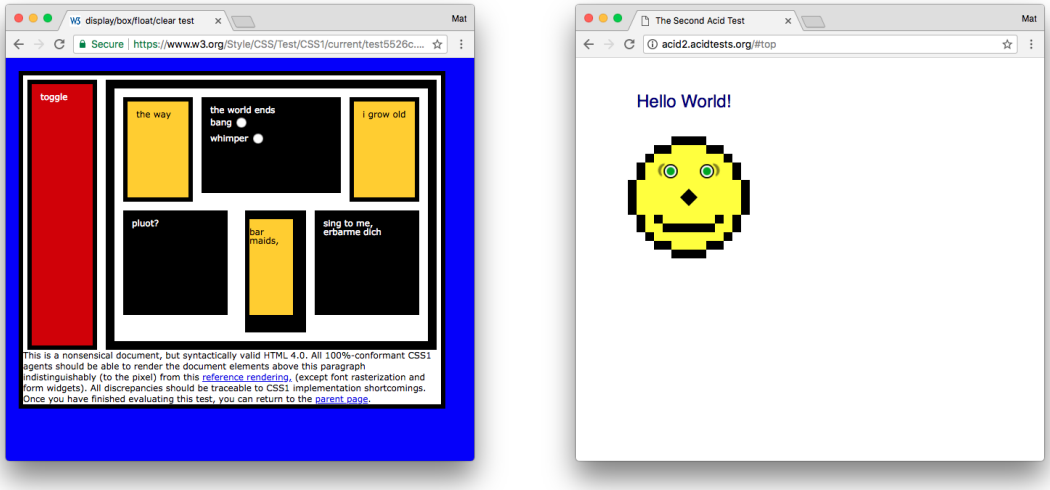
We created a list of resources referenced in the HTML tags and CSS. The difference between the total set of resources loaded and the resources referenced in the HTML and CSS are assumed to come from JavaScript. We found that the the CC measure is directly related to the number of JavaScript requests to external resources. By taking the average across all environments, we found that the Twitter set resources load 16.3% of the requisite resources through JavaScript (presumably

Ajax), whereas 18.7% of resources are loaded via JavaScript in the Archive-It set. This was contrary to our hypothesis that increased CC will produce more resource requests from JavaScript. The Twitter set, which has more embedded JavaScript (CC = 4.78), makes fewer requests to content with JavaScript than the seemingly less complex Archive-It set (CC = 2.16).

4.2.3 ARCHIVAL ACID TEST

Because archival crawlers attempt to duplicate what a user would see if he accessed the page on the live Web, variance from what is preserved and what would have been seen compromises the integrity of the archive. The functional difference between archival crawlers and Web browsers causes this sort of unavoidable discrepancy in the archives, but it is difficult to evaluate how good of a job the crawler did if the information no longer exists on the live Web. By examining what sort of Web content is inaccurately represented or missing from the Web archives, it would be useful to evaluate the capability of archival crawlers (in respect to that of Web browsers that implement the latest technologies) to determine what might be missing from their functional repertoire.

Web browsers exhibited this deviation between each other in the early days of Web Standards. A series of "Acid Tests" that implemented the Web Standards allowed each browser to visually and functionally render a Web page and produce an evaluation of how well the browser conformed to the standards (Figure 41). In much the same way, we created an "Archival Acid Test" [105] to implement features of Web browsers in a Web page. While all standards-compliant browsers will correctly render the live page, this is not always the case when the archived version of the page is rendered. This difference can be used to highlight the features that archival crawlers are lacking compared to Web browsers and thus emphasize the deviations that will occur in Web archives compared to what a user would expect from a digitally preserved Web page.



(a) Acid1 Test **(b)** Acid2 Test

Fig. 41: Acid Tests were a means of testing Web browser conformance to Web Standards based on how a page was rendered as compared to a reference image. We adapted this model for the Archival Acid Test [105] to evaluate the quality of the capture of various Web archiving tools and services. A third iteration, the Acid3 Test, is displayed in Figure 42.

Inspired by the Acid Tests administered by the Web Standard Project (WaSP)[2], we built the Archival Acid Test[3] to evaluate how well archival tools of 2014 (when the study was completed) perform at preserving Web pages. Unlike WaSP's initiatives, evaluation of Web archival software is not standardized, so a comprehensive test of what these tools should be able to capture needs to be established. The Archival Acid Test evaluates the archives' ability to re-render pages employing a variety of standardized and emerging conventions with HTML and JavaScript.

The crux of the tests was to determine how well an archival tool preserves a Web page in terms of similarity to what would be expected by a user viewing the page from the live Web, i.e., a respectively modern Web browser. Web Standards are continuously evolving with the feature set for Web browsers temporally lagging the standards in being implemented though frequently containing experimental implementations. Archival crawlers, given a greater need for reliability, lag in implementing newly

[1] `http://www.w3.org/2004/12/rules-ws/paper/44/`
[2] `http://www.acidtests.org/`
[3] The source of the test is available at `https://github.com/machawk1/archivalAcidTest`

standardized features as compared to browsers, though they will frequently rely on a common engine utilized by browsers to stay-up-to-date.[4] The deviation from the Web page processing engines used by archival tools (whether built-to-purpose or older versions of browser engines) is a source of discrepancy between the content on a live Web page and that which is captured by these tools.

We established a set of tests into three categories to better group Web page features that might be problematic for archival tools to capture. Each test was represented by a 10-by-10 pixel blue square. Any deviation from the blue square (e.g., no image present, red square instead of blue) signifies an error in what a user would expect from a preserved Web page, and thus the particular test is considered to have been failed by the tool. A reference image (Figure 43a) is used as a comparative basis for correctness, much in the same way Web Standards Acid Tests provided a static image to evaluate what was experienced versus what is right.

Basic Tests (Group 1)

The set of *Basic Tests* is meant to ensure that simple representations of resources on Web pages are captured. Each tests' name represents what is presented to be captured by the archival crawler. A sample URI follows each test's name.

1a. Local (same server as test) image, relative URI to test

`./1a.png`

1b. Local image, absolute URI

`http://archiveacidtest/1b.png`

1c. Remote image, absolute URI

`http://anotherserver/1c.png`

1d. Inline content, encoded image

`...`

1e. Remote image, scheme-less URI

`//anotherserver/1e.png`

1f. Recursively included CSS

In style.css: `@import url("1f.css");`

[4]For example, the open source V8 and SpiderMonkey rendering engines allow resources that require JavaScript to be present on a Web page and be captured by archival tools.

JavaScript Tests (Group 2)

The second group of tests is meant to evaluate the archival crawler's JavaScript support in terms of how the script would execute were the test accessed on the live Web with a browser.

2a. Local script, relative URI, loads local resource

```
<script src="local.js" />
```

2b. Remote script, absolute URI, loads local resource

```
<script src="http://anotherserver/local.js" />
```

2c. Inline script, manipulates DOM[5] at runtime

```
<script>...(JS code)...</script>
```

2d. Inline script, Ajax image replacement, loads local resource

```
img.src = "incorrect.png";
...code to replace incorrect image with local...
```

2e. Inline script, Ajax image replacement, Same-origin Policy (SOP)[6] enforcement, replacement (bad) == false positive

```
img.src = "correct.png"';
...code to replace correct image with image
from SOP violation...
```

2f. Inline script, manipulates DOM after delay

```
setTimeout(function(){ ...load image...},2000);
```

2g. Inline script, content loaded upon interaction, introducing resources

```
window.onscroll = function()
```

2h. Inline script, add local CSS at runtime

Advanced Features Tests (Group 3)

The third group of tests evaluates script-related features of HTML beyond simple DOM manipulation.

[5]Document Object Model, the structure of the Web page that, when manipulated, affects the content

[6]https://developer.mozilla.org/en-US/docs/Web/JavaScript/Same_origin_policy_for_JavaScript

3a. HTML5 Canvas drawing with runtime-fetched content

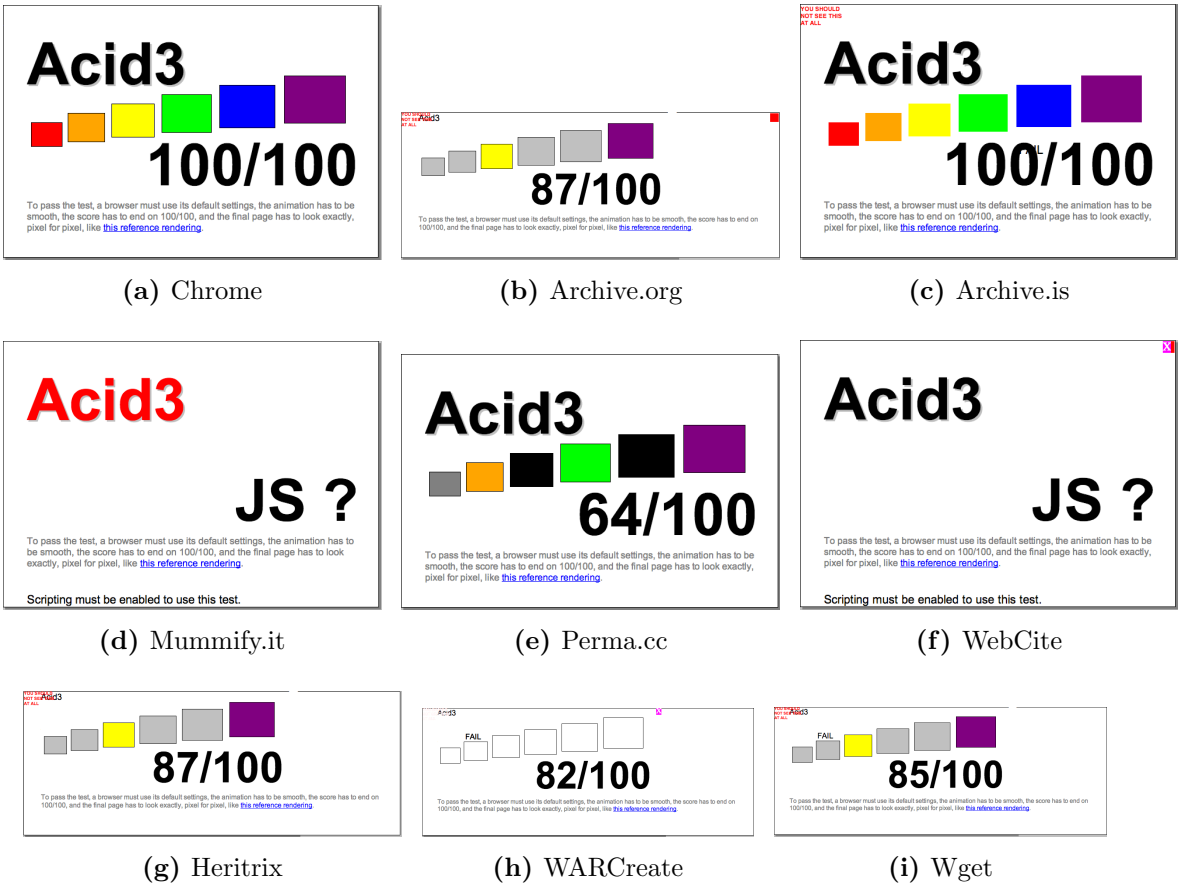
3b. Remote image stored then retrieved from HTML5 localStorage

3c. Embedded content using iframe

3d. Runtime binary object

Evaluation

To establish a baseline, we first ran each tool through the Acid3 test. From this we observed preliminary results that were indicative of the archival tools' lack of full support of the features of standards compliant Web browsers (Figure 42). Given that we are testing features that have come about since Acid3 was released, the Archival Acid Test further exercised the tested sites' and tools' standards compliance and specifically highlights their failures.



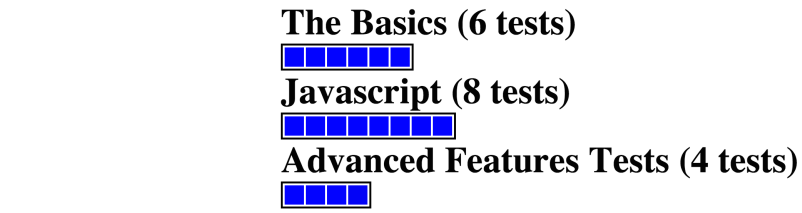
(a) Chrome **(b)** Archive.org **(c)** Archive.is

(d) Mummify.it **(e)** Perma.cc **(f)** WebCite

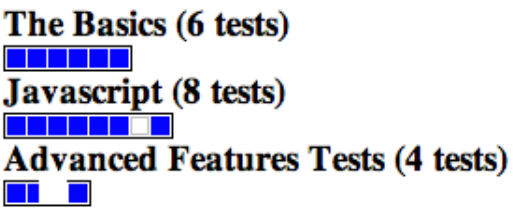
(g) Heritrix **(h)** WARCreate **(i)** Wget

Fig. 42: Preliminary tests show that archival tools exhibit an incomplete feature set compared to modern Web browsers. Tests run in January 2014.

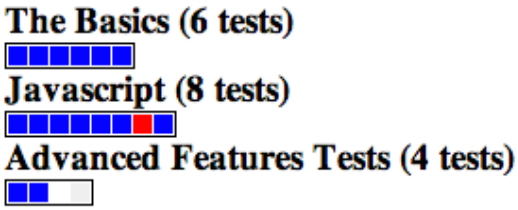
In Figure 42, we show the results of each tool's attempt at capturing the Acid3 Test Web page. Compared to the correct rendering in Chrome (Figure 42a), the five service-based tools from Archive.org, Archive.is, Mummify.it, Perma.cc, and WebCite (Figures 42b, 42c, 42d, 42e, and 42f, respectively) have more variance in their performance than the three tools of Heritrix, WARCreate, and Wget (Figures 42g, 42h, and 42i, respectively). While Archive.is appears to get the closest with its rendering, subtle stylistic differences are easily observable with error text appearing. This indicates that contrary to the 100/100 rating, neither Archive.is nor any other service or tool tested here fully passes the Acid3 test.



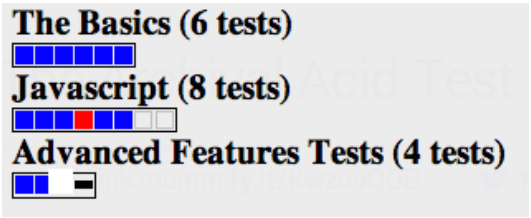
(a) Chrome



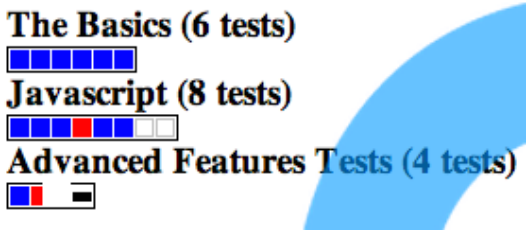
(b) Archive.org



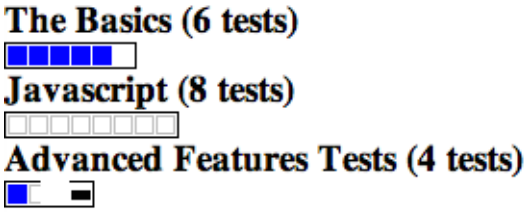
(c) Archive.is



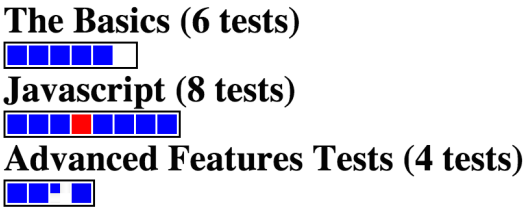
(d) Mummify.it



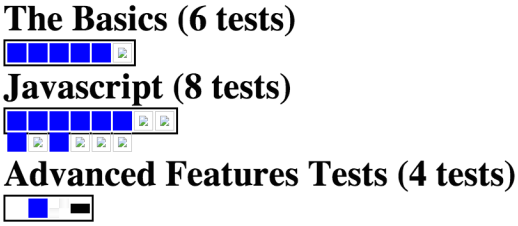
(e) Perma.cc



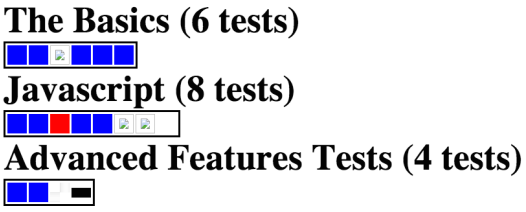
(f) WebCite



(g) Heritrix



(h) WARCreate



(i) Wget

Fig. 43: Archiving service and tools' performance on the Archival Acid Test. The reference image shows what should be displayed if all tests are passed. This image represents what a user sees when viewing the test in a modern Web browser. Tests run in January 2014.

Tool \Test	1a	1b	1c	1d	1e	1f	2a	2b	2c	2d	2e	2f	2g	2h	3a	3b	3c	3d
Archive.org	·	·	·	·	·	·	·	·	·	·	·	·	✗	·	·	·	✗	·
Archive.is	·	·	·	·	·	·	·	·	·	·	·	·	✗	·	·	·	✗	✗
Mummify.it	·	·	·	·	·	·	·	·	·	✗	·	·	✗	✗	·	·	✗	✗
Perma.cc	·	·	·	·	·	·	·	·	·	✗	·	·	✗	✗	·	✗	✗	✗
WebCite	·	·	·	·	·	✗	✗	✗	✗	✗	✗	✗	✗	✗	·	✗	✗	✗
Heritrix	·	·	·	·	·	✗	·	·	·	✗	·	·	·	·	·	·	✗	·
WARCreate	·	·	·	·	·	✗	·	·	·	·	·	·	✗	✗	✗	·	✗	✗
Wget	·	·	✗	·	·	·	·	·	✗	·	·	✗	✗	✗	·	·	✗	✗

· = Test Passed ✗ = Test Failed

TABLE IV: By aligning the services' and tools' tests and failures, a theme in capability (and lack thereof) is easily observable between the two classes. Where archiving services exhibit a perfect record in the Group 1 set, the Group 2 set proved troublesome for all but Heritrix. Further, the nearly across-the-board failures of 2g and 3c when modern browsers pass all of the tests emphasizes the functional discrepancy between archiving tools and browsers.

Tools' Performance

We evaluated five Web archiving services (Archive.org, Archive.is, Mummify.it, Perma.cc, and WebCite) and three WARC-generating archiving tools (Heritrix, WARCreate, and Wget). Each service provided a simple interface where a user can submit a URI, and the Web page at that URI is preserved on-command. Heritrix was configured with the test as the lone URI in a crawl. Wget was given a command[7] with the URI as a parameter and WARC as the desired output format. For WARCreate, we navigated to the test's Web page and generated a WARC. For each WARC-generating archiving tool, we replayed the generated WARC files in a local instance of Wayback[8].

While almost all archiving services and tools tested had difficulty with test 2g, the five service-based archiving Web sites (Archive.org, Archive.is, Mummify.it, Perma.cc,

[7] `wget --mirror --page-requisites --warc-file="wget.warc" http://{acid test URI}`

[8] OpenWayback version 2.0.0BETA2, the latest SNAPSHOT, built from source

and WebCite Figures 43b, 43c, 43d, 43e, and 43f, respectively) show an interesting common set of features compared to the three archiving tools (Heritrix, WARCreate, and Wget, Figure 43g, 43h, and 43i, respectively). This is better illustrated in Table IV.

The features of the Archival Acid Test are not necessarily bleeding edge, yet no service or tool completely passed. More advanced features were considered but as a preliminary test of evaluating the targets, the 18 tests presented in the Archival Acid Test were more than sufficient at pointing out their shortcomings. Of particular interest are tests 2g and 3c, which tested whether the targets were able to capture content loaded after a short delay and content embedded in an iframe. In one of our previous experiments [100], we evaluated content already in the archives that existed in frames, so this discrepancy was unexpected.

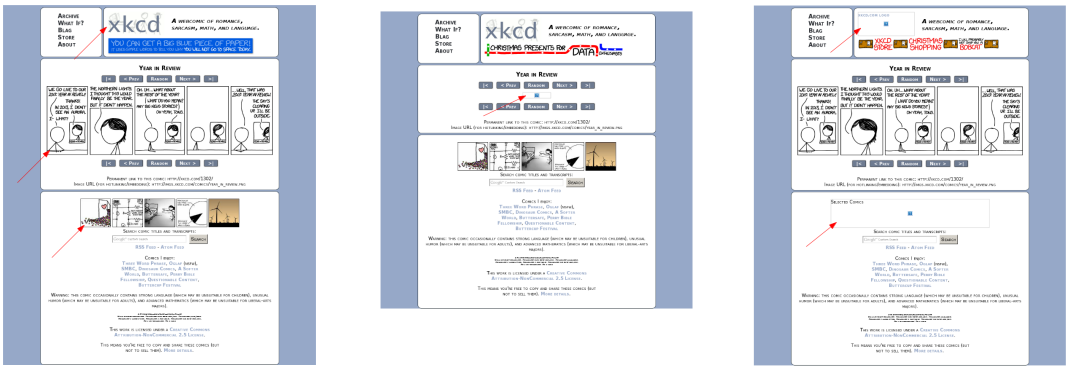
4.2.4 MEMENTO DAMAGE

Researchers have studied the completeness of the archives, the re-crawl policies that optimize archive quality, and the relative importance of content within Web resources. To extend on our previous work studying the factors that influence archivability [100], we developed a metric called "Memento Damage" [46, 47] to quantify the measure as experienced through replay. Previous studies by Reyes et al. [20] found that this process was almost always performed manually. In our work, we sought to understand how missing embedded resources impact Web users' perceived quality of a memento. Using an algorithm to measure embedded resource importance, we determined whether an important embedded resource of the memento is missing, or if the missing embedded resource contributes little to the memento's utility for the user. Using a method for weighting embedded resources according to importance (D_m), we showed that D_m was an improved damage rating over an unweighted proportion of missing embedded resources (M_m) for all requested resources. We evaluated our algorithm in three ways:

- Using Amazon Mechanical Turk to compare our algorithm to Web users' notion of damage and to show an improvement over the unweighted count of missing resources.
- Applied our algorithm to assess damage in Internet Archive and WebCite to compare D_m to M_m variance at scale.

- Applied our algorithm on IA and WebCite to show how damage has changed over time.

We initially used the XKCD Web page as an example of a resource with embedded resources of differing importance. After capturing the URI-R[9] and manually inflicting damage by removing images, we evaluated the captured URI-M, captured a screenshot, and recorded the HTTP response headers of the embedded resources. From there, we created three mementos of the URI-R: one duplicating the live Web (m_0, Figure 44a) with verification of a damage value equivalent to the URI-R[10], one with the central comic image removed (m_1, Figure 44b), and one with two logo images removed (m_2, Figure 44c) but the main comic image intact. Where it is visually apparent that the main comic image is of greater importance to the page's understanding, the effects of a single missing resource may be much more catastrophic.



(a) m_0 replicating the live Web, all images intact

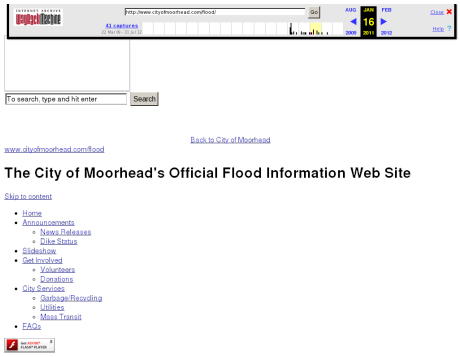
(b) m_1 with main image removed

(c) m_2 with logos removed

Fig. 44: We created three mementos of XKCD. In two of the three, we removed select images to evaluate resource importance.

[9] `http://www.xkcd.com`

[10] The live Web page was missing two embedded stylesheets. This characteristic was inherently propagated to each memento.



(a) In some Web pages, a single missing CSS file can have a drastic visual effect on rendering...

(b) ...while in other captures, the missing stylesheets are not visually apparent.

Fig. 45: A missing stylesheet completely changes the appearance of a memento (a) where in other cases (like (b) and Figure 44), a missing stylesheet had no apparent effect on the memento's rendering.

Archivists' perceptions of damage differ from those of traditional Web users. To determine whether M_m (percent missing) is a good estimate of human perception of damage, we used Amazon's Mechanical Turk to measure human agreement with M_m. We presented "turkers" with pairs of mementos that had varying levels of damage and asked them to select the memento they preferred to keep if given a choice between the two.

Prior to defining equations for our memento quality measurements, we first describe the resources in the mementos in Equation 4, differentiating between the set of all embedded resources R and the set of all missing resources R_m. In this case, we consider any resource needed to build a resource that is requested by the client an *embedded resource*. R is calculated by counting the number of distinct and unique URIs requested by the client when dereferencing the URI-M. Our measurement of M_m (Equation 5) is the proportion of missing embedded resources to all requested resources. Defining M_m as a proportion normalizes the measurement for pages that have a very large or very small number of embedded resources. From this we can calculate the cumulative damage D_m (Equation 6) of a memento m as a normalized value ranging from $[0, 1]$. We calculate the potential damage of a memento and the actual damage of a memento and express the damage rating as the ratio of actual to

potential damage.

$$\begin{aligned} R &= \{\text{All embedded resources requested}\} \\ R_m &= \{\text{All missing embedded resources}\} \\ R_m &\subseteq R \end{aligned} \tag{4}$$

$$M_m = \frac{R_m}{R} \tag{5}$$

We captured and manually damaged 11 hand-selected URI-Rs using five different methods of damage: missing image, missing CSS, missing all images, missing all resources, and original. Each permutation of damage calculated from URI-R-to-method is displayed in Table V.

$$D_m = \frac{D_{m_{actual}}}{D_{m_{potential}}} \tag{6}$$

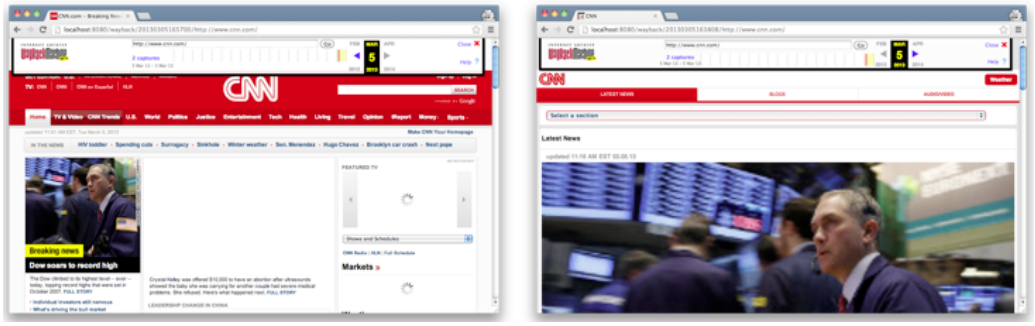
URI-R	m_0	M_m missing image	missing css	missing all images	missing all
http://www.cs.odu.edu/~mln/	0.14	0.43	0.29	0.43	0.43
http://activehistory.ca/2013/06/myspace-is-cool-again-too-bad-they-destroyed-history-along-the-way/comment-page-1/	0.00	0.32	0.32	0.57	0.85
http://www.albop.com/	0.00	0.13	0.00	0.50	0.50
http://www.cs.odu.edu/	0.10	0.13	0.11	0.82	0.81
http://ws-dl.blogspot.com/2013/08/2013-07-26-web-archiving-and-digital.html	0.07	0.08	0.08	0.13	0.14
http://www.cnn.com/2013/08/19/tech/social-media/zuckerberg-facebook-hack/	0.19	0.22	0.28	0.46	0.57
http://xkcd.com/	0.14	0.38	0.31	0.53	0.54
http://www.mozilla.org/	0.80	0.80	0.80	0.88	0.89
http://www.ehow.com/	0.05	0.05	0.06	0.11	0.33
http://google.com/	0.00	0.00	0.00	0.00	1.00
http://php.net/	0.32	0.33	0.33	0.37	0.37

TABLE V: The 11 URI-Rs used to create the manually damaged dataset. M_m values are provided for each m_1.

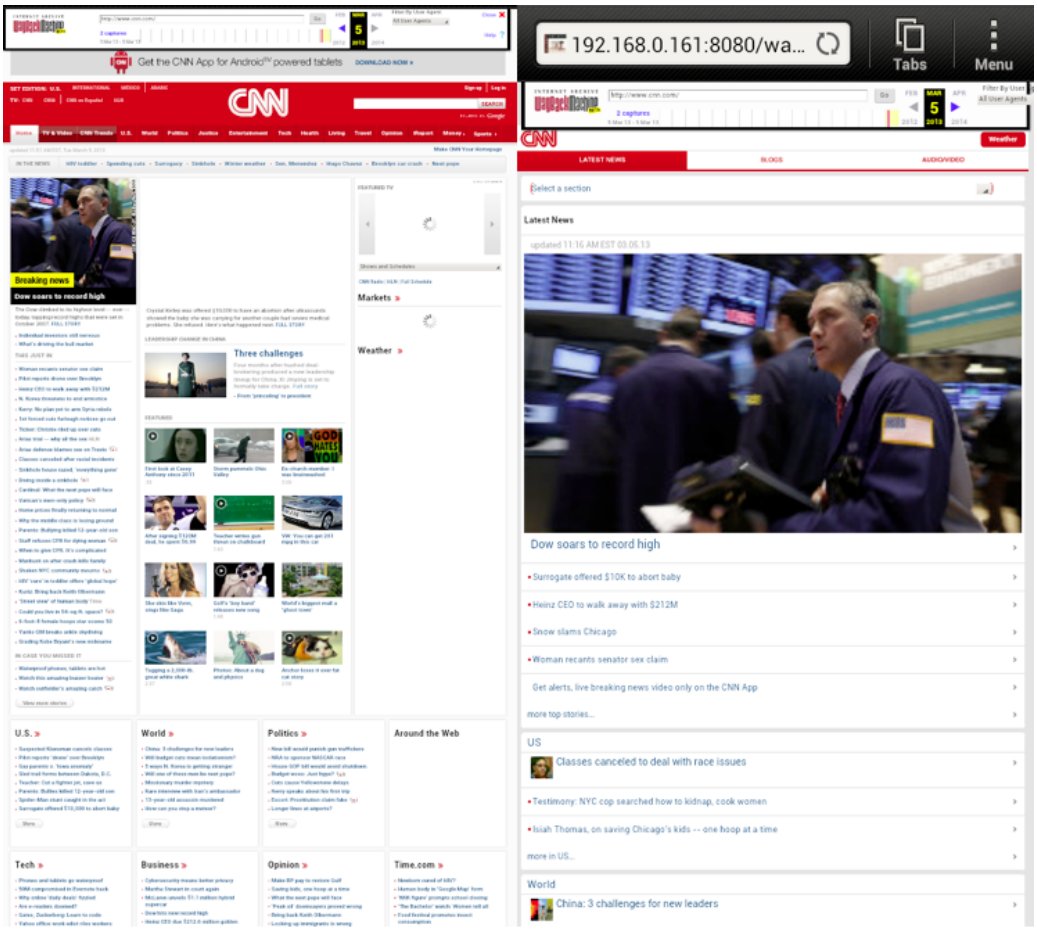
4.2.5 IDENTIFYING PERSONALIZED REPRESENTATIONS

Aggregating personal and institutional captures requires being able to identify personalization of captured content, both at the time of capture and potentially after-the-fact. We performed an analysis using a locally hosted OpenWayback within WAIL and multiple sources of WARC files to allow a replay system to identify personalized representations at the time of replay [99]. The personalization we detected was the source user-agent, location of capture (GeoIP), etc. that we could then show an option in the OpenWayback interface (Figure 47) to change perspectives based on the perspective used to view the capture (e.g., mobile vs. desktop; viewing from

Portsmouth, Virginia or Washington, DC; see Figure 46).



(a) Heritrix uses a desktop Mozilla user-agent for capture, accessed at replay from a Mac (left) and uses a iPhone Mozilla user-agent for capture, accessed at replayed from a Mac (right).



(b) Heritrix uses a desktop Mozilla user-agent for capture, accessed from an Android phone (left) and uses an iPhone Mozilla user-agent for capture, accessed at replay from an Android phone (right).

Fig. 46: The user-agent string specified by the crawler when preserving the page and the string used by the user-agent to view the replay of a capture affects the displayed result in the viewport.

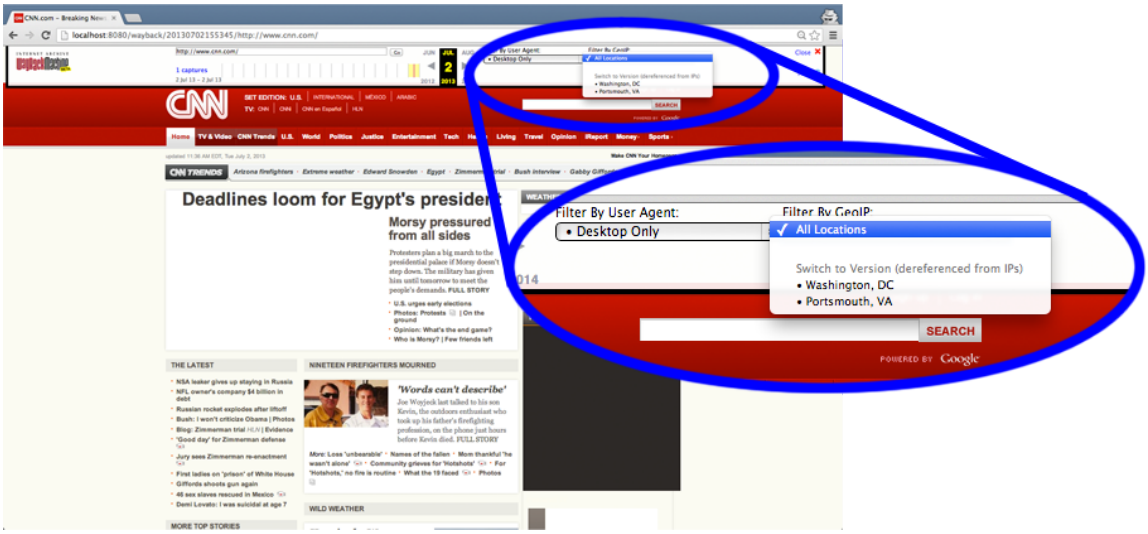


Fig. 47: We modified the OpenWayback replay system to allow traversal of captures that meet the criteria of a selected additional dimension, e.g., only captures from a location.

4.3 PEER-TO-PEER COLLABORATION AND PROPAGATION

We developed InterPlanetary Wayback (ipwb) [95, 6] to propagate Web archive content into the peer-to-peer InterPlanetary File System (IPFS) to promote sharing of archived content and mitigate efforts that otherwise result in duplication. With private Web archivists being the target audience for this software, IPFS allows a rudimentary level of access control and encryption that we address further in this proposal. Upon the extracted contents of WARC files being added to IPFS via ipwb (Figure 48), a unique content-derived hash is generated that allows the file to be retrieved in IPFS by this hash. Our initial prototype retained these hashes and associated them with a URI-R and datetime combination via CDXJ (Section 2.4.4) for retrieval (example ipwb CDXJ shown in Figure 49). For a user to share WARC files, the user adds the relevant contents to IPFS using ipwb, obtains the hashes, then shares the individual hashes or generated CDXJ file. This process can be performed en-masse to propagate Web archives to facilitate preservation through redundancy.

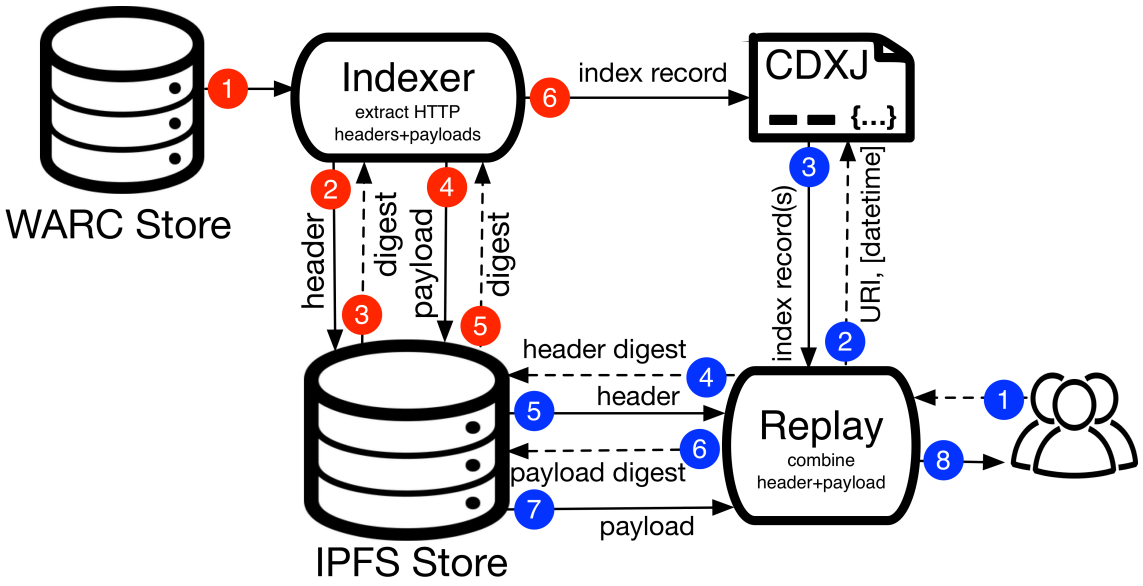


Fig. 48: Pushing WARC records to IPFS (red circles) requires the WARC response headers and payloads to be extracted (red 1), pushed to IPFS to obtain digest hashes (red 2-5), and hashes to be included in an index (red 6). The replay process (blue circles) has a user querying a replay system as usual (blue 1) that obtains a digest for the URI-datetime key from the index (blue 2 and 3), which is used as the basis for retrieving the content associated with the digests from IPFS (blue 4-7). The replay system can then process these payloads as if they were in local WARC files and return the content to the user (blue 8).

Content may be retrieved from IPFS, even without ipwb, using the content addressed hash representative of the content itself. In the context of Web archiving and particularly private Web archiving, this may be problematic, as content in WARCs from private Web archive may contain sensitive or personally identifiable information. In an initial effort [101], we extended ipwb to allow for encryption of the content extracted from a WARC at time of dissemination. A generated CDXJ will then contain the associative entries with the IPFS hashes being representative of the encrypted WARC contents within IPFS. A user that intercepts the CDXJ file, in this case, must decrypt the content at the hash. This rudimentary approach is to be expanded in this proposal, as it serves as one mechanism for securely propagating WARCs and may serve as a cornerstone in collection building of Web archives through collaboration.

```
SURT_URI DATETIME {
        "id": "WARC-Record-ID",
        "url": "ORIGINAL_URI",
        "status": "3-DIGIT_HTTP_STATUS",
        "mime": "Content-Type",
        "locator": "urn:ipfs/HEADER_DIGEST/PAYLOAD_DIGEST"
}
```

Fig. 49: A CDXJ index allows a memento to be resolved to a WARC record in a playback system. In the ipwb prototype we extract the relevant values from the HTTP response headers at time of index and include the IPFS hashes as the means for a replay system to obtain the HTTP headers and payload corresponding to the URI-M requested.

4.4 SUMMARY

In this chapter we described preliminary work and research performed to lead up to the development of the Mementity Framework, to be described in detail in the next chapter. Section 4.1 described our work in building tools to enable individuals to preserve contents on the live Web with browser-based and familiarly interfaced tools. In Section 4.2 we described our investigations of measuring the archivability of resources on the live Web and quantified how this has changed over time, as reflected in Web archives. In Section 4.3 we described our work in integrating distributed file systems and addressing with Web archives with the intention of facilitating collaboration of Web archives through secure propagation of personal and private captures.

CHAPTER 5

PROPOSED FRAMEWORK

> *It's hard enough getting people to share data as it is, harder to get them to share it in a particular format, and completely impossible to get them to store it and manage it in a completely new system.*

- Aaron Swartz, *Aaron Swartz's A Programmable Web, An Unfinished Work [163]*

In this chapter we describe a framework (henceforth the "Mementity Framework") for aggregating private and public Web archives based on the state of the art in Web archiving and the preliminary research described in Chapter 4. This chapter addresses work to be completed to answer Research Questions 4-6:

> **RQ4:** How can content that was captured behind authentication signal to Web archive replay systems that it requires special handling?
>
> **RQ5:** How can Memento aggregators indicate that private Web archive content requires special handling to be replayed, despite being aggregated with publicly available Web archive content?
>
> **RQ6:** What kinds of access control do users who create private Web archives need to regulate access to their archives?

The Mementity Framework will provide the constructs and methodologies for the aggregation of private, public, and personal Web archives. The target archives shall be aggregated systematically to account for access restrictions and to allow collaboration and sharing of personal and private archival Web captures. "Aggregation" is minimally executed as a list of identifiers (e.g., URI-Ms) and associated attributes (e.g., datetime) in a Memento TimeMap. The primary goal of this chapter is to lay out the approach to be used to answer Research Questions 4-6, all which deal with access to Web archives, with the introduction of additional *mementities* in the Web archiving workflow that resolve these questions. The term "mementity" in this work correlates with the role of traditional Memento aggregators and TimeGates in

conventional usage. In this proposal we introduce three mementities: the Memento Meta-Aggregator, the Private Web Archive Adapter, and the StarGate.

With the introduction of the mementities into a Web archiving workflow, we will enable the capabilities and mitigate the shortcomings of a conventional workflow, as described in Chapter 1. For instance, preserving content behind authentication (e.g., bank statements and time-limited verification documents, as in Section 1.2) may require a degree of access control and negotiation in dimensions beyond time, as provided by the Private Web Archive Adapter and StarGate mementities, discussed in Sections 5.2.2 and 5.2.3, respectively. A means of sharing captures but controlling who can see and access the captures (per the scenarios in Section 1.3) may be enabled by a combination of the Private Web Archive Adapter mementity and Memento Meta-Aggregator mementity (described in Section 5.2.1). The hierarchical and role-based nature of each mementity is designed to be interoperable, extensible, and applicable to a variety of currently existing Web archiving use cases. Introduction of the mementities also enables the investigation and abilities required to answer all six research questions.

The state of the art of conventional Memento aggregation is exhibited between multiple public Web archives. Until recently, Memento aggregators at institutions (like the one hosted at mementoweb.org[1]) served as the primary and sole method for end-users to obtain aggregated Memento TimeMaps and perform multi-archive temporal negotiation. The creation of an open source, easily configurable, locally hosted Memento aggregator (MemGator [9], Section 3.1) removes the barriers of enabling aggregation of a custom set of archives. A locally hosted aggregator also facilitates further research for the necessary considerations of aggregating personal and private Web archives, as described in this research. The Mementity Framework supplements results from conventional aggregators to produce a TimeMap that may contain identifiers (URI-Ms) and associated attributes for captures from private Web archives, captures of public content from private archives (e.g., a user's `cnn.com` captures), and Memento-compliant public Web archives when dereferenced.

The aggregation of personal, private, and public URI-Ms necessitates consideration of content negotiation with archives in dimensions other than time, as provided by Memento. In Section 5.1, we outline negotiation of this sort in the context of how to express these additional dimensions in the conventional TimeMap medium

[1] e.g., `http://timetravel.mementoweb.org/timemap/link/http://matkelly.com`

and the potential origin of an initial sample set of these derivatives. The Mementity Framework requires three additional mementities in the hierarchy of accessing Web archives. The scope of each mementity as a precursor to the role a mementity plays in the Mementity Framework is described in Section 5.2. Introducing mementities into a Web archiving workflow provides an extensible and interoperable approach with new abstract and concrete capabilities like new methods of negotiation, archival precedence (Section 5.3.2), and potential for inter-archive and archive-to-user collaboration, as discussed in Section 5.3. Section 5.4 describes a preliminary set of User Access Patterns that we initially anticipate and relates how each pattern corresponds to the scenarios and research questions described in Chapter 1, where applicable. Section 5.5 discusses the extensibility of the Mementity Framework both in the role of the mementities in Web archival dynamics and to account for unanticipated access models, allowing the framework be extended to currently unforeseeable Web archiving scenarios.

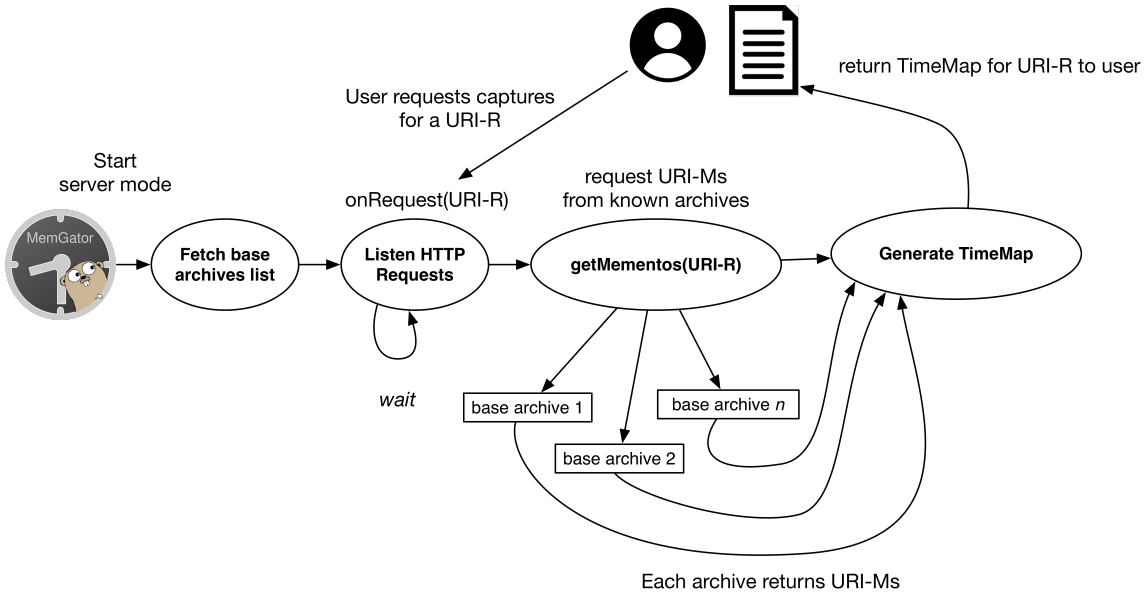
5.1 ARCHIVAL NEGOTIATION BEYOND TIME

Memento TimeMaps do not currently provide standard syntax for representing additional arbitrary attributes about the captures they describe. These attributes may be defined using a variety of methods. For example, the mementity generating the TimeMap may refer to an external Web service to obtain values for a set of URI-Ms, the attribute values may be calculated by the client (e.g., subjective quality evaluation using a computational means like Web Workers), etc. Allowing for the amendment of TimeMaps with yet-to-be-defined attributes allows for the approach to be agnostic of a specific means and extensible to other unforeseeable methods. One barrier in preventing TimeMaps from being more expressive, as previously described, is the Link format that is defined in RFC 5988 [130] (on which the Memento Framework was based), its obsoleting successor RFC 8288 [132], and CoRE [156] syntax. In a more recent solution designed specifically with Web archives in mind, Alam et al. [5, 10] defined the CDXJ format (Section 2.4.4), an extension of the conventional CDX [83] archival indexing format, as an extensible means of associating additional attributes to URI-Ms in both the context of archival indexes and allowing TimeMaps to be more expressive and semantically extensible. As discussed in Section 2.4.4, CDX files serve as indexes for Web archive files and contain many fields (e.g., MIME-type, status code, and content-digest of the memento) that are

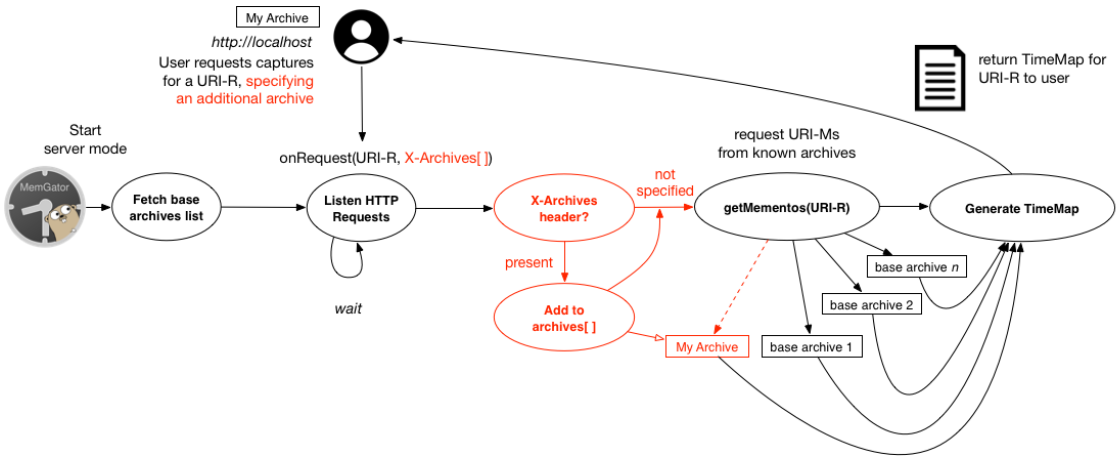
not present in TimeMaps. In cases where the basis for generating a TimeMap is an archival index (e.g., CDX listing), these attributes are readily available for inclusion but currently are not expressed in TimeMaps for a URI-R. In other scenarios, additional attributes may need to be calculated prior to being expressed in TimeMaps. In this proposal we distinguish TimeMaps with additional attributes for mementos beyond URI-M and datetime as "StarMaps" where "star" is an allusion to "*" to indicate a wildcard of dimensions beyond time. A TimeMap describing only the conventional attributes of Memento is considered a StarMap with no additional attributes.

Clients that access Web archives often do so by requesting a URI-R and datetime and being returned the closest URI-M, requesting a URI-M directly, or requesting a URI-T to get a list of captures. It is not common practice for a client to have more sophisticated interaction with Web archives as they would on the live Web. For instance, Memento aggregators are not currently receptive to a client specifying the set of archives used as the basis for aggregation. In a more sophisticated scenario, clients are not currently able to request a TimeMap with characteristics or formal attributes that meet a specified criteria. For example, a user may wish to only obtain the `nasa.gov` or `cnn.com` mementos that contain damage [47] under a certain threshold (Figures 6 and 7, respectively). In this section we explore ways to resolve these outstanding issues by investigating archival negotiation beyond time. In Section 5.1.1 we investigate different mechanisms using existing standards for client-side specification of the set of archives aggregated by a Memento aggregator. In Section 5.1.2 we discuss different categories of attributes for URI-Ms that would enrich TimeMaps to make them more useful and descriptive of the archives' holdings. In Section 5.1.3 we provide a high-level description on how attributes that require inspection of the memento itself may be acquired and expressed. Upon defining the initial description of archival negotiation beyond time in this section, we may then, in the following section, discuss the Mementity Framework further in the context of mementities themselves and how they enable these additional negotiation constructs and dynamics.

5.1.1 CLIENT-SIDE ARCHIVAL SPECIFICATION



(a) MemGator server mode.



(b) User specification of additional archives in the MemGator archival processing flow (changes highlighted in red) allow for implementation of the precedence and short-circuiting model (Section 5.3.2).

Fig. 50: MemGator conventionally works on a predefined set of archives initialized on startup. By enabling clients to modify the set of archives at runtime, users can effectively aggregate additional archives of their choosing through specification of an archive's attributes through an extended MemGator's HTTP endpoints.

MemGator [9] is an open source Memento aggregator that supports CDXJ TimeMaps (Section 2.5) in addition to conventional Link and JSON formatted TimeMaps. In this work, we will adapt the code for MemGator to effectively serve as an implementation of a mementity to handle additional HTTP request parameters supplied by a client as well as to produce TimeMaps with the additionally proposed attributes.

Much like a conventional Memento aggregator, MemGator works with a static set of Web archives initially set upon starting the server process (Figure 50a). A feature in adapting MemGator is to allow interaction of the set of archives from which to build the aggregated TimeMap as well as how to interact with each archive. We are initially investigating the merits of using one of three different approaches to allow for client-side archival specification using:

1. A separate HTTP request header, e.g., `X-Archives` (Figure 50b)
2. The Prefer HTTP request header [160] with encoded JSON [41] (Figure 53)
3. A client-modified, server-supplied Cookie-based [23] approach

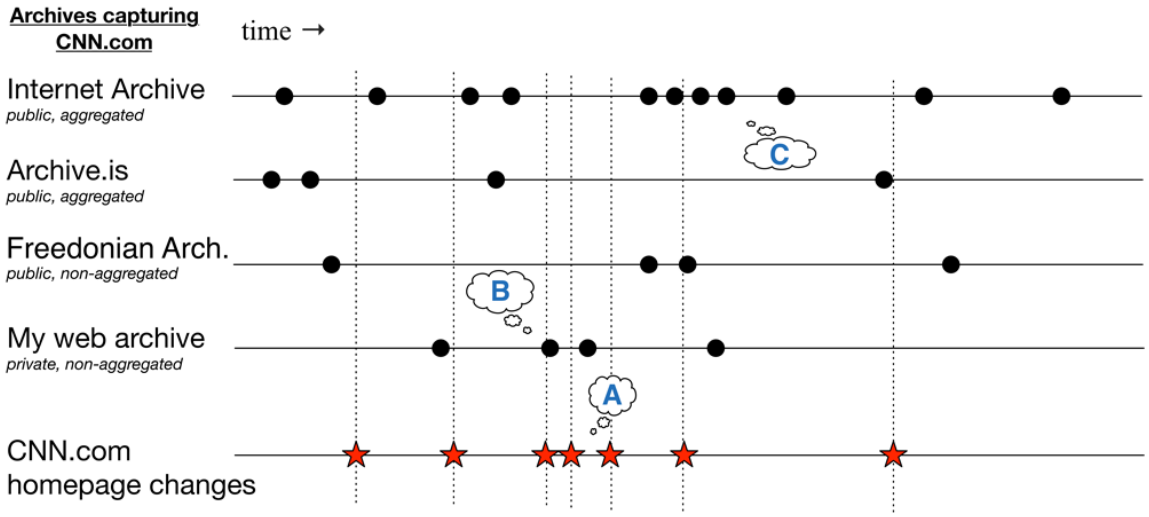


Fig. 51: Personal Web archives allow captures from institutional archives to be supplemented. For a URI-R (e.g., `cnn.com`) that changes frequently (marker A), a Web scale archive may only preserve the page after multiple representations have occurred (marker C). Aggregation of captures with personal and private Web archives would allow these missing representations (marker B) to show a more temporally comprehensive picture (or one with more accurate replay per Figure 7) of how the page has changed over time.

Providing the ability for a user to interact with an aggregator by providing the identifiers for archival supplementation is novel. Users often act as "pure clients" to these services in that, beyond requesting results for a URI-R, they have no say as to the sources to query for this URI-R. The purpose of allowing a client to specify a custom set of archives to an aggregator is to not necessarily permanently affect the

sources of the aggregator but rather, allow the aggregator to perform the aggregation process with a custom set of archival sources. For example, a user may provide additional archival sources to an aggregator to result in a more comprehensive picture of how a `cnn.com` news story evolved using an increased temporal snapshot rate facilitated with the introduction of additional sources (Figure 51). Conversely, a user may wish to aggregate captures from a completely disjoint set of archives than provided by an aggregator but still leverage the aggregator's capability. For example, a user may wish to exclude archives that only preserved login pages of `facebook.com` while referring only to a specified set of personal captures (Section 1.2). While the ramifications of providing private sources to public Web services are accounted for by the various mementities defined in this work, the capability of providing a completely custom set of sources may be useful in other scenarios. In preliminary research (Section 4.1.4), we provided a graphical means for a user to query an aggregator using their Web browser and a browser extension, Mink. The initial capability of Mink did not allow for the user to customize the set of archives aggregated. Figure 52 describes a mockup of Mink that would allow client-side archival specification were the Mementity Framework in-place and aggregators receptive to client-side archival specification.

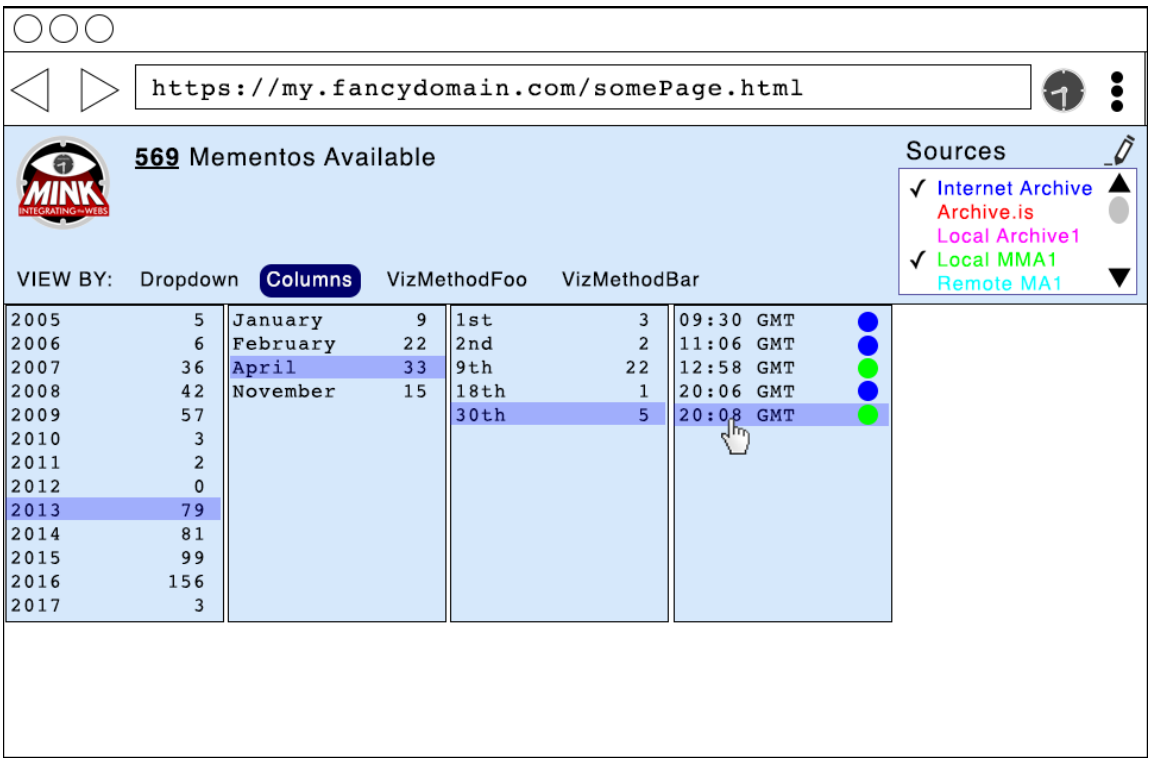


Fig. 52: Mink could potentially be extended to allow a client to specify the set of archives aggregated. However, unless Mink is itself performing the aggregation, the aggregator must understand the semantics and syntax of client-side archival specification.

Allowing a user to specify a custom set of Web archives through an HTTP header prior to an aggregator commencing archival communication allows for the potential of integrating more sophisticated querying models (like precedence and short-circuiting of requests to archives, discussed in Section 5.3.2). In a preliminary prototype[2], we extended MemGator to simply allow a client to provide more archival endpoints to be taken into account at runtime using an `X-Archives` HTTP request header. This simplification provided a base proof-of-concept of modifying the set of archives aggregated without the potential scenarios of requesting a subset, supplemented intersecting set (base plus additional), or a disjoint set. Despite the naïve and ad hoc approach (and the deprecation of "X-" prefixed headers [153]), using this simplistic means of archival specification allowed the declaration of additional archives to be in clear text, which makes end-user customization easier. However, the approach is not scalable, semantically expressive, or standard, so we opted to investigate more standards-driven method for client-side archival specification.

[2] `https://github.com/machawk1/gogator/`

(a) **(b)**

```
[
  {
    "timemap": "http://web.archive.org/web/timemap/link/",
    "timegate": "http://web.archive.org/web/"
  },
    {
    "timemap":
    ↪ "http://www.webarchive.org.uk/wayback/archive/timemap/link/",
    "timegate": "http://www.webarchive.org.uk/wayback/archive/"
  },
  {
    "timemap": "http://archive.today/timemap/",
    "timegate": "http://archive.today/timegate/"
  }
]
```

(c)

```
[
    {
    "timemap":
    ↪ "http://www.webarchive.org.uk/wayback/archive/timemap/link/",
    "timegate": "http://www.webarchive.org.uk/wayback/archive/"
  },
  {
    "timemap": "http://freedonia.archive/timemap/",
    "timegate": "http://freedonia.archive/timegate/",
  }
]
```

(d)

Fig. 53: Using a Prefer-based archival supplementing model, a user may request the list of archives from an aggregator (a) then submit her own set (b) using the format. Here, she receives a configuration with three archives from the aggregator (c) and specifies a set of two (d) with only a single archive being contained within the intersection of the set provided and the set supplied.

A second approach we are investigating to allow client-side archival specification is by using the Prefer header for HTTP Prefer [160]. Figure 53 shows an example where a user requests the list of archives from an aggregator (Figure 53a) and receives a JSON payload containing three archives (Figure 53c). This approach assumes a more capable and transparent aggregator than is curently offered. After receiving the list of supported archives ($\{A_0\}$), a client may use the Prefer header to construct their own JSON-formatted list of archives, matching the format that the archive specified. Using this method would allow a user to amend the list of archives with additional archives, supplement a subset of the supported archives, or provide an entirely disjoint set of archives to use as sources for aggregation. The JSON block (for example, if an aggregator provides this format) may be transmitted using the syntax of Prefer with the payload encoded for transmission, e.g., `Prefer: archives="data:application/json;charset=utf-8;base64,Ww0KI...NCn0="`. Figure 53 shows Carol using the format of archival specification provided by the aggregator to construct a JSON file containing the specification of two archives (Figure 53d) and submitting this back to the aggregator (Figure 53b) to be applied onto subsequent requests.

```
> GET /timemap/link/http://fox.cs.vt.edu/wadl2017.html
↪ HTTP/1.1
> Host: mma.cs.odu.edu
> Prefer: archives="data:application/json;charset=utf-8;
↪ base64,Ww0KICB7...NCn0="

< HTTP/1.1 200
< content-type: application/link-format
< vary: prefer
< preference-applied: return=representation;
↪ archives="data:application/json;charset=utf-8;
↪ base64,Ww0KICB7...NCn0="
< content-location:
↪ /timemap/link/5bd...8e9/http://fox.cs.vt.edu/wadl2017.html
```

Fig. 54: Client-side specification of a set of archives via encoded JSON using HTTP Prefer. The Memento aggregator responds with the location of a TimeMap for the URI-R at a URI-T representative of the set.

More sophisticated aggregation may require filtering on a memento-level (e.g., only source mementos from archives with a certain quality of capture) or on a TimeMap-level. Memento TimeGate allow for datetime resolution but not server-side filtering of the results prior to returning a response. For instance, a user may wish to provide a previously unaggregated public archive (e.g., the "Freedonia Web Archive" in Figure 53b) or a private/personal Web archive as an additional source for aggregation. A conventional Memento aggregator may be required to provide additional parameters or communication flows to obtain mementos for a URI-R from private Web archives. In the current operation, a Memento aggregator assumes that all archives in a set are willing to provide a TimeMap in all instances without further parameters needing to be specified. This may not be the case for a client's personal archive or a public Web archive that is not currently included in the aggregated set.

We anticipate a 3-step process for a client to specify the archive set:

1. Client requests the set of archives to be aggregated by default from a Prefer-aware Memento aggregator (Figure 53a).

2. The aggregator returns the set of archives, e.g., as a JSON (per MemGator) or an XML (per `mementoweb.org`) file (Figure 53a), represented as $\{A_0\}$.

3. Once a response is received from the aggregator (e.g., `https://git.io/archives`), a client may manipulate the contents to be either an identical set ($\{A_f\} = \{A_0\}$), subset ($\{A_f\} \subset \{A_0\}$), supplementary set ($\{A_f\} \supset \{A_0\}$), or disjoint set ($\{A_f\} \dot{\bigcup} \{A_0\}$) (Figure 53b) and submit back to the aggregator for subsequent queries (Figure 54).

A client may also manipulate an existing archive's specification in the response received. For instance, a profiling probability (a value already defined in the MemGator specification) may be manipulated or a value of query precedence or short-circuiting may be modified.

Given that no Memento aggregator yet supports the client-side archive specification, we extend this idea with the assumption that a JSON response is received (like MemGator and Webrecorder's aggregator). A client may perform step 3 using the HTTP Prefer request header. After potentially manipulating the JSON response, a client would encode the JSON as a base64-encoded data URI (or supply some other URI for specification-by-reference) and submit a request with the Prefer header and a URI-R (Figure 54).

5.1.2 ENRICHING TIMEMAPS TO PRODUCE STARMAPS

Memento TimeMaps may conventionally contain URI-Ms (for mementos), URI-Gs (for TimeGates), URI-Ts (for TimeMaps) and associative relation types (e.g., `original`, `timemap`, `next`) for each identifier. We initially anticipate and here describe three new types of attributes for richer TimeMaps: **content-based attributes** based on data when dereferenced, **derived attributes** requiring further analysis beyond dereferencing but useful for evaluating capture quality, and **access attributes** that guide users and software as to requirements needed to dereference mementos in private archives, personal archives, and archives with access restrictions. We refer to TimeMaps containing these additional attributes beyond time as StarMaps. These more expressive attributes will guide us in answering the research question relating to how aggregators can indicate content that requires special handling when dereferenced for both replay and Memento-style aggregation (RQ4). We refer to TimeMaps containing these additional attributes beyond time as StarMaps. This section details

the enrichment of TimeMaps to produce StarMaps.

Content-based Attributes

Determining how many mementos exist from an archive for a URI-R is impossible from a TimeMap alone [98]. Enriching a TimeMap with information about the dereferenced captures would improve methods for determining how well (both potentially in quantity and quality) a URI-R has been captured. HTTP data obtained when dereferencing a URI-M, like status code [67], content-type [67], and Last-Modified, are often used to gain information about archival holdings without requiring each URI-M be repeatedly dereferenced. It is likely that not all Web archives will report values for some or all URI-Ms for a URI-R due to irrelevancy or lack of support. The loose nature of JSON objects allows for this inter-record imbalance, i.e., some URI-Ms may have a particular attribute assigned (even those from the same archive) while other do not.

Derived Attributes

Researchers often analyze the contents of a memento and generate derived data from this analysis. For example, Brunelle et al. [47] developed a metric for determining the quality of a capture (cf. content-based attributes) when dereferencing a URI-M with a particular focus on the quantitative significance of missing embedded resources. Determining "Memento Damage" requires calculation beyond simple counting, as all resources are not equally weighted in importance, particularly when absent. Having this information calculated and present in a TimeMap would allow a user to select the best or most complete URI-M without needing to iterate through URI-Ms.

As a follow-on to the discussion on content-based attributes, a hash or content-digest of the archived payload would allow selecting unique captures that are not redirects much easier. In previous work [95, 6], we explored using content addressing to facilitate de-duplication of content in Web archives. Despite some Web archives providing an endpoint to obtain CDX records for a URI-M that provides content-digest, many Web archives do not provide such an endpoint. As this data is often easily calculated upon accessing the content using standard hashing mechanisms (Internet Archive uses `sha1` base-32), the result could be retained and used for subsequent TimeMaps for the URI-R.

For identifying significant changes in a Web page over time, AlSum and Nelson [15] applied the SimHash [52] algorithm, which requires comprehensive asynchronous generation of a value for all mementos followed by synchronous offline calculation of Hamming distance. They then used the Hamming distance values as the basis for selection for which URI-Ms to generate thumbnails as a representative summary of a URI-R over time. The bulk of the latency for the thumbnail summarization procedure, despite being asynchronous, resides in initially generating SimHashes for all mementos from the URI-Ms in a TimeMap. Retaining the SimHash values once calculated and supplying them in TimeMaps alongside corresponding URI-Ms would allow the synchronous operation to be performed on-demand.

Both Memento Damage and SimHash are examples of computationally expensive operations for a URI-R. Retaining these values and expressing them in TimeMaps for a URI-R would save users of TimeMaps from having to regenerate data based off of these and other of derived attributes.

Access Attributes

A goal of this research is to provide a framework for aggregating private, personal, and public Web archives by using and extending Memento. To provide access control for select mementos, we require a means to specify access-related attributes. The final space-delimited field in the CDXJ format (Figures 57) consists of a JSON block with a minimal but extensible set of JSON object attributes. The CDXJ format's JSON field allows additional attributes to be specified and considered when a URI-M is dereferenced. To express access attributes that are based on neither the contents nor derived values from the contents of a memento, we leverage the encapsulating and associative nature of the JSON block in CDXJ; that is, attributes of a URI-M may be nested to describe more scope-specific detail (Figure 59).

Standard authentication procedures as used on the live Web will help to inform our design decisions of applying the practice to the archived Web. Take the scenario where a blog allows a user to log in using their Facebook credentials. The blog wishes to allow the user to post as their identity, so upon clicking a button, redirects a user to a `facebook.com` address to explicitly authorize the access. There, a user may log in using the Facebook authentication system and in turn, Facebook provides a unique token to the user to be returned and relayed to the blog's commenting system. This unique token prevents the user's credentials from being required by the blog.

Upon posting with this associative token, the blog can then reuse this token to obtain additional information about the user (e.g., their name) to populate the comment metadata.

Mapping this model to accessing private Web archives, at time of access to a private Web archive, the user will be redirected by the archive to a different memento for authentication. After authentication using a similar method to the live Web, the user can use the token to access private Web archives captures as configured by the authentication memento and the archive itself. This relationship need not be boolean, for example, if the token imposes bounds to the set of URI-Ms, URI-Rs, time range, or any combination of these or additional characteristics as configured by the archive.

Access control may be needed in cases where private and personal Web archives are aggregated with public Web archives via StarMaps. An authentication procedure and subsequent tokenization will allow persistent access using a token derived from authenticating. A token may be attributed on the basis of a particular URI-M (the token is valid only for that capture), or all URI-Ms from that archive (potentially defined in the CDXJ metadata for brevity). For example, Figure 56 shows a potential simplified workflow of a user gaining access to a private Web archive. In this scenario, the archive is aware of the requirement for further credentials to authorize access, so it redirects the user to a memento at a second location to obtain this. Upon obtaining the credentials from the user, the second memento returns a token attributed to a URI-M and the user's credentials. The user may then use this token along with the original URI-M to then gain access to the URI-M in the private Web archive.

```
19981212013921 {
  "uri": "http://localhost:8080/20101116060516/http://facebook.com/",
  "rel": "memento",
  "datetime": "Tue, 16 Nov 2010 06:05:16 GMT",
  "status_code": 200,
  "digest": "sha1:LK26DRRQJ4WATC6LBVF3B3Z4P2CP5ZZ7",
  "damage": 0.24,
  "simhash": "6551110622422153488",
  "content-language": "en-US",
  "access": {
    "type": "Blake2b",
    "token":
    ↪ "c6ed419e74907d220c69858614d8669ff3732df0cc5647ef0a3a88a41..."
  }
}
```

Fig. 55: An amended CDXJ record for a private capture of `facebook.com`. Line breaks added for readability.

The responsibility for attributing the token to an individual or set of mementos may lie in either the archive itself or from the aggregator. Figure 55 shows an example enriched CDXJ record containing attributes describing how the token is stored in an enriched CDXJ TimeMap (a StarMap). The example uses OAuth2 [74] for authorization when dereferencing URI-Ms with this field and the BLAKE2 hashing algorithm [152] for tokenization for persistent access to private mementos.

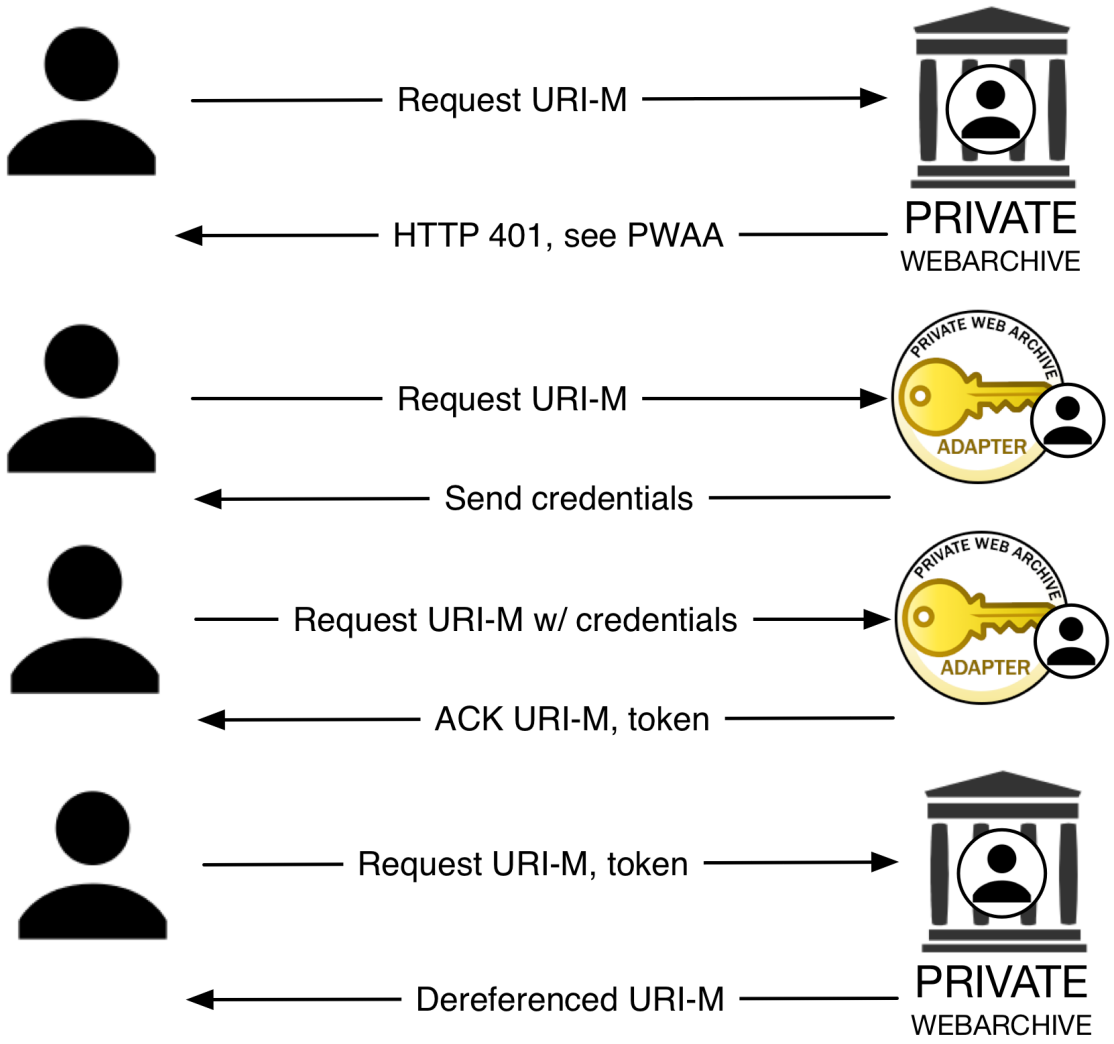


Fig. 56: A private Web archive may deny anonymous access to its contents, potentially reporting an HTTP 401 even if it contains no captures for a URI-R. The archive should then refer the client to a Private Web Archive Adapter to authenticate and obtain a token that can then be used to request the contents of the private Web archive.

5.1.3 SOURCES OF DERIVED ATTRIBUTES

CDXJ allows metadata fields (lines beginning with `@meta`) about the TimeMap to precede the listing of captures. Figure 57 contains metadata fields (highlighted in red) within a CDXJ TimeMap that are typically also found in a Link-formatted TimeMap (Figure 58), e.g., URI-R for the original resource, TimeGates, other related TimeMaps, etc. With the introduction of derived attributes (Section 5.1.2), it is

critical to not just give context as to the semantics of new attributes like “damage” but also to provide guidance in generating this value.

```
!context ["http://tools.ietf.org/html/rfc7089"]
!id {"uri": "http://localhost:1208/timemap/cdxj/http://facebook.com"}
!keys ["memento_datetime_YYYYMMDDhhmmss"]
!meta {"original_uri": "http://facebook.com"}
!meta {"timegate_uri":
↪ "http://localhost:1208/timegate/http://facebook.com"}
!meta {"timemap_uri": {"link_format":
↪ "http://localhost:1208/timemap/link/http://facebook.com",
↪ "json_format":
↪ "http://localhost:1208/timemap/json/http://facebook.com",
↪ "cdxj_format":
↪ "http://localhost:1208/timemap/cdxj/http://facebook.com"}}
19981212013921 {"uri":
↪ "http://archive.is/19981212013921/http://facebook.com/", "rel":
↪ "first memento", "datetime": "Sat, 12 Dec 1998 01:39:21 GMT"}
19981212013921 {"uri":
↪ "http://web.archive.org/web/19981212013921/http://facebook.com/",
↪ "rel": "memento", "datetime": "Sat, 12 Dec 1998 01:39:21 GMT"}
19981212024839 {"uri":
↪ "http://web.archive.org/web/19981212024839/http://www.facebook.com/",
↪ "rel": "memento", "datetime": "Sat, 12 Dec 1998 02:48:39 GMT"}
...
20170330231113 {"uri":
↪ "http://web.archive.org/web/20170330231113/http://www.facebook.com/",
↪ "rel": "memento", "datetime": "Thu, 30 Mar 2017 23:11:13 GMT"}
20170331013527 {"uri":
↪ "http://web.archive.org/web/20170331013527/https://www.facebook.com/",
↪ "rel": "last memento", "datetime": "Fri, 31 Mar 2017 01:35:27 GMT"}
```

Fig. 57: An abbreviated CDXJ TimeMap from MemGator for `facebook.com`. Metadata records highlighted in red.

```
<http://facebook.com>; rel="original",
<http://localhost:1208/timemap/link/http://facebook.com>; rel="self";
↪ type="application/link-format",
<http://archive.is/19981212013921/http://facebook.com/>; rel="first
↪ memento"; datetime="Sat, 12 Dec 1998 01:39:21 GMT",
<http://web.archive.org/web/19981212013921/http://facebook.com/>;
↪ rel="memento"; datetime="Sat, 12 Dec 1998 01:39:21 GMT",
<http://web.archive.org/web/19981212024839/http://facebook.com/>;
↪ rel="memento"; datetime="Sat, 12 Dec 1998 02:48:39 GMT",
...
<http://web.archive.org/web/20170330231113/http://facebook.com/>;
↪ rel="memento"; datetime="Thu, 30 Mar 2017 23:11:13 GMT",
<http://web.archive.org/web/20170331013527/http://facebook.com/>;
↪ rel="last memento"; datetime="Fri, 31 Mar 2017 01:35:27 GMT"
<http://localhost:1208/timemap/link/http://facebook.com>;
↪ rel="timemap"; type="application/link-format",
<http://localhost:1208/timemap/json/http://facebook.com>;
↪ rel="timemap"; type="application/json",
<http://localhost:1208/timemap/cdxj/http://facebook.com>;
↪ rel="timemap"; type="application/cdxj+ors",
<http://localhost:1208/timegate/http://facebook.com>; rel="timegate"
```

Fig. 58: An abbreviated Link TimeMap from MemGator for `facebook.com`.

Figure 59 provides an example where a derived attribute requiring calculation (memento damage [47]) and an access attribute are defined for guidance within the StarMap. Definitions in the `extended_attributes` metadata field serve as templates as applied to URI-Ms in the StarMap when present and applicable. The "service" `rel` value (inspired by Atom [73]), for instance, instructs parsers to look to the URI specified in the template and the URI-M itself to obtain a value for this attribute. The "access" attribute is given a contextual definition using a `rel` value of "self via" [134] wherein the expectation is for parser to look to the URI-M where the access attribute exists for resolution. The "via" `rel` value [134] for each of these attributes instructs parsers to look to the respective identifier for the source of the information for the links context: "self" for the access attribute, and "service" for the service defined by the URI for the JSON block.

```
@meta {"extended_attributes": {
  "damage": {rel="service via", "service":
  ↪ "http://memento-damage.cs.odu.edu/?uri={uri}",
  ↪ type="float"},
  "access": {rel="self via", "token": "self",
  ↪ type="string"}
}}
```

Fig. 59: Additional metadata atop a StarMap provides guidance to both the user and generation tools to produce derived attributes for URI-Ms in a TimeMap.

5.2 MEMENTITIES

In this section we define three functional Mementities and their role as part of the makeup of the Mementity Framework:

Memento Meta-Aggregators (MMAs)
Archival aggregation with considerations beyond public Web archives

Private Web Archive Adapters (PWAAs)
Access regulation to private and personal Web archives

StarGates (SGs)
Content negotiation with Web archives in dimensions beyond time

Reference implementations for each mementity will be provided as software to serve the respective role of their purpose in the Mementity Framework. Extensive details about each mementity are provided in Sections 5.2.1, 5.2.2, and 5.2.3, respectively.

5.2.1 MEMENTO META-AGGREGATOR

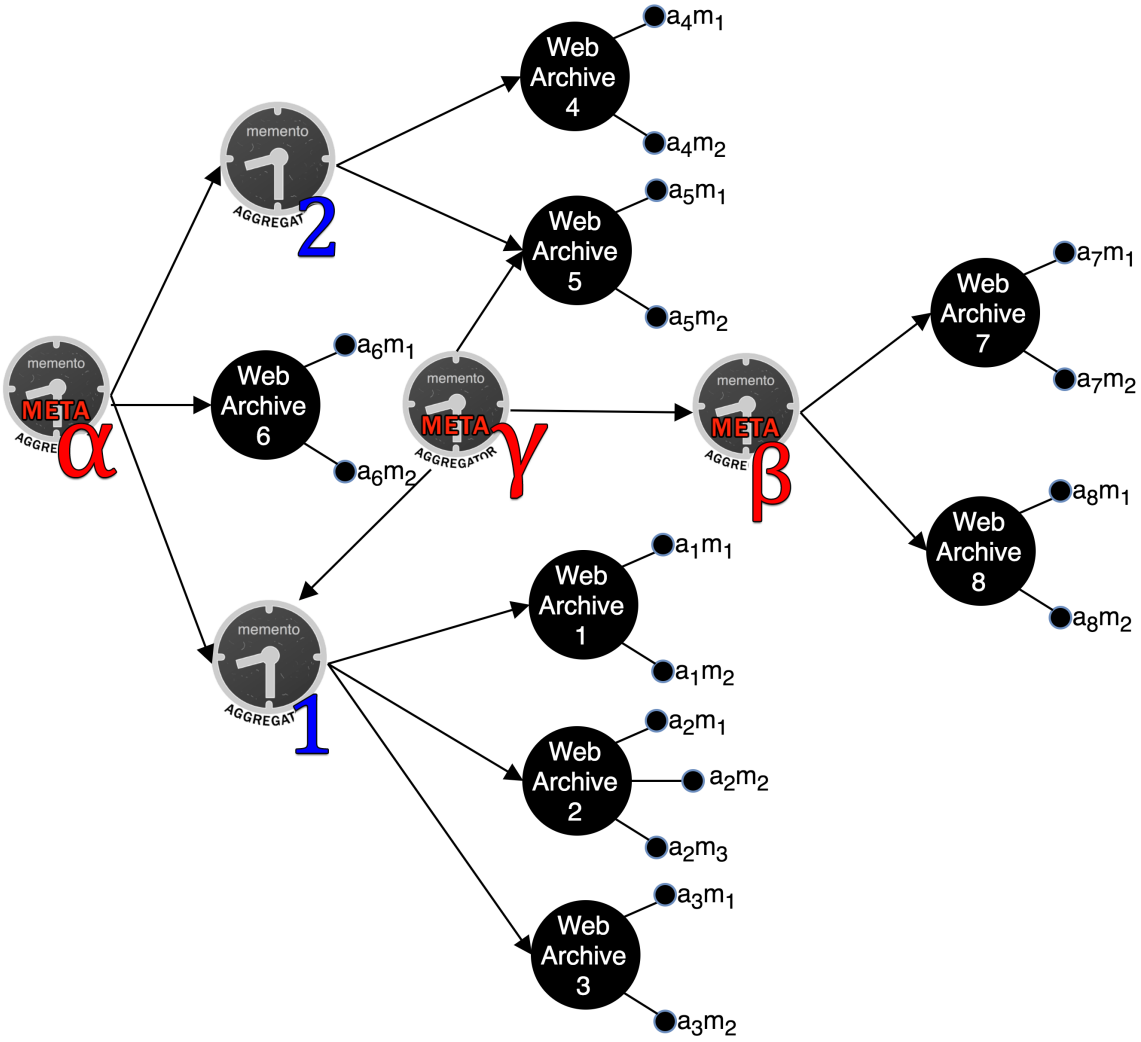
Memento Aggregators typically combine URI-Ms from the results of querying multiple Web archives. A Memento Meta-Aggregator (MMA) serves as a functional superset of a conventional Memento Aggregator (MA), along with adding functionality outside of the scope of a conventional MA. A conventional MA provides access through identifiers to mementos (URI-Ms), TimeGates (URI-Gs), and TimeMaps

(URI-Ts) from a set of Web archives. An MMA provides the ability to both supplement and selectively filter the results returned from an MA with URI-Ms from additional Web archives at the request of the user or as configured with the MMA. Results from other Web archives that are aggregated with the results from an MA may be public non-aggregated Memento-compliant Web archives or private Web archives as relayed through a Private Web Archive Adapter (Section 5.2.2). A conventional MA is not required for the functionality of an MMA. An MMA may serve as a functional replacement for an MA at a fundamental level; that is, the aggregation of a static set of public Web archives may be performed by an MMA in a black box manner as if the MMA were identically configured with the same archives as the MA.

Figure 60 describes a sample hierarchical relationship of mementities consisting of MMAs, MAs, and Web archives (WAs). When MA_1 receives a request for URI-Ms for a URI-R, for instance, the request is relayed to WA_1, WA_2, and WA_3 for the sets of mementos $\{a_1m_1, a_1m_2\}$, $\{a_2m_1, a_2m_2, a_2m_3\}$, and $\{a_3m_1, a_3m_2\}$, respectively. MA_1 is then responsible for combining and temporally sorting the URI-Ms then returning the aggregated StarMap to the requesting user (or mementity). The temporal ordering within an archive corresponds to the second index (m) for convenience in the figure, however, this ordering may not hold between archives. For example, a_2m_2 is older than a_3m_1 per the temporal ordering diagram in Figure 61a. The ordering for the mementos contained within the configured archives as requested from various mementities is displayed in Figure 61b. This figure also shows examples of an MMA obtaining results from multiple MAs (e.g., MMA_α from MA_1 and MA_2) and even MMAs referring to other MMAs for their results when queried (e.g., MMA_γ referring to MA_1, WA_5, and MMA_β with the latter referring to WA_7 and WA_8). The configuration of MMA_β is similar to the relationship of MMA_{Carol} to MMA_{Alice} in Figure 62 where a user may configure an MMA to both refer to a custom set of sources for results as well as reuse the in-place selective filtering of the sources. In this case, MMA_{Carol} would inherit the restriction of MMA_{Alice} of not sending requests for mementos of `http://alicesembarassingphotos.net/vacation.html` to Bob's archive.

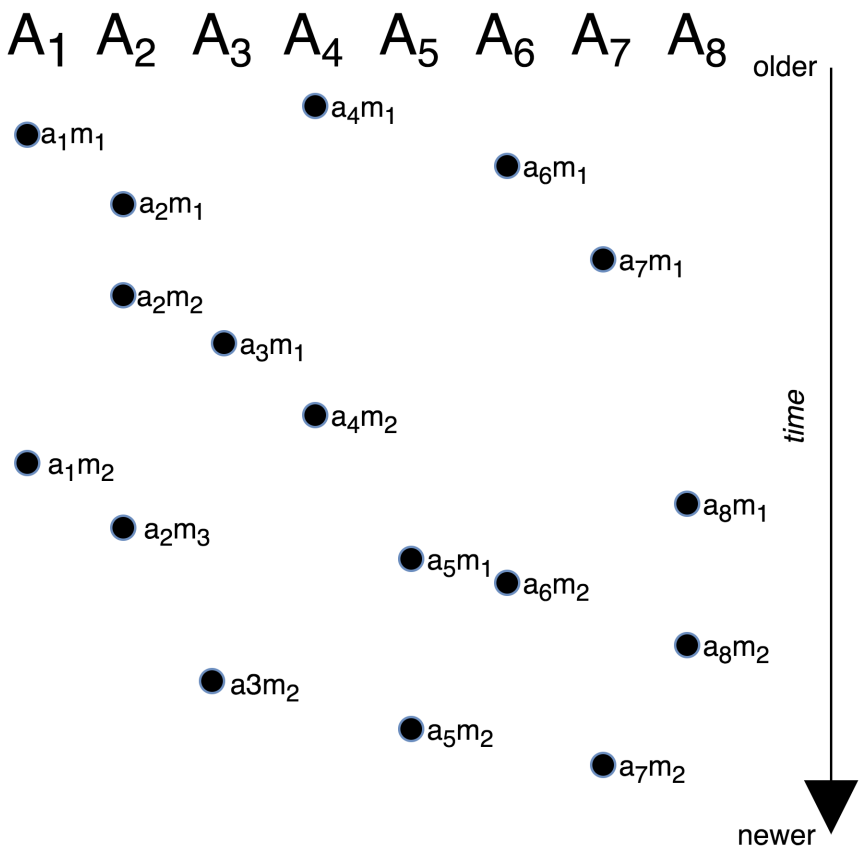
An MMA can be configured to return an aggregated StarMap based on a set of Web archives for which it has been configured or provided a set of archives to query upon request. This abstraction provides a level of extensibility to current Memento aggregators for which the additional functionality may not be appropriate, scalable, or interoperable, however, providing an on-demand set of archives to query is useful

in the context of personal Web archiving.



$A_{1...n}$	Archive 1 of n
$MA_{1...n}$	Memento Aggregator 1 of n (denoted using integers)
$MMA_{\alpha...\omega}$	Memento Meta-Aggregator 1 of n (denoted using Greek)
● $a_x m_y$	Memento of index y from archive of index x
Ⓐ⟶Ⓑ	Mementity A fetches results from mementity B

Fig. 60: Memento Meta-Aggregators may aggregate URI-Ms from archives, Memento aggregators, and other MMAs equivalently. Shown is an example of temporally sorted captures as served from an MMA in a variety of permutations in a potentially ad hoc hierarchy. The temporal ordering and mementos aggregated by each mementity are described further in Figure 61.



(a) Temporal ordering of mementos aggregated in the hierarchy described in Figure 60.

Mementity	**→Abstracted Holdings**	**→Memento Holdings**
MA_1	$\{A_1, A_2, A_3\}$	$\{a_1m_1, a_2m_1, a_2m_2, a_3m_1, a_1m_2, a_2m_3, m_3m_2\}$
MA_2	$\{A_4, A_5\}$	$\{a_4m1, a_4m2, a_5m_1, a_5m_2\}$
MMA_α	$\{MA_1, MA_2, A_6\} \rightarrow$ $\hookrightarrow \{A_1, A_2, A_3, A_4, A_5, A_6\}$	$\{a_4m_1, a_1m_1, a_6m_1, a_2m_1, a_2m_2, a_3m_1, a_4m_2,$ $\hookrightarrow a_1m_2, a_2m_3, a_5m_1, a_6m_2, a_3m_2, a_5m_2\}$
MMA_β	$\{A_7, A_8\}$	$\{a_7m_1, a_8m_1, a_8m_2, a_7m_2\}$
MMA_γ	$\{MA_1, A_5, MMA_\beta\} \rightarrow$ $\hookrightarrow \{A_1, A_2, A_3, A_5, A_7, A_8\}$	$\{a_1m_1, a_2m_1, a_7m_1, a_2m_2, a_3m_1, a_1m_2, a_8m_1,$ $\hookrightarrow a_2m_3, a_5m_1, a_8m_2, a_3m_2, a_5m_2, a_7m_2\}$

(b) The set of mementos aggregated depends on the set of the abstracted holdings, which may be an archive, another Memento Aggregator, or a Memento Meta-Aggregator

Fig. 61: The temporal ordering of URI-Ms in a StarMap depends on the set of archives aggregated in a StarMap. Per Figure 60, the set of archives aggregated by each mementity determines the set of mementos returned.

User-driven specification of aggregation parameters is particularly important for

accessing personal Web archives using a Memento aggregator. If a user requests a TimeMap from a conventional Memento aggregator, the aggregator will request the URI-Ms for the URI-R from each archive with which the aggregator is configured to communicate. A user may wish to customize, prioritize, or give precedence to the archives queried. If a user were to host an aggregator themselves, the aggregator would need to be reconfigured to prevent requests for the URI-R from propagating to certain archives on the basis of {URI-R, archive} pairs. Though this may become unwieldy, what follows is a useful example to illustrate where configuring an MMA with a core ruleset prior to considering further user-driven specification would be useful when aggregating personal and public Web archives.

Consider the scenario where Alice archives Web pages she views in her browser using WARCreate [107], and replays them using her local Wayback instance within WAIL [102] (Section 4.1). Bob, who is Alice's acquaintance, and Carol, who is Alice's sister, each do the same for their own captures. Alice sets up a Memento Meta-Aggregator that is configured to request captures from her archive, Bob's archive, Carol's archive, and the Internet Archive. For some URIs, like `facebook.com` it may not make sense to aggregate Alice, Bob, and Carol's captures with those from Internet Archive (see the example in Figure 10)[3]. For other URIs, Alice may want to prevent exposing to Bob and the Internet Archive the fact that she is looking for certain old captures (as inferred by an aggregator sending a request for mementos for a URI-R), but wants to also aggregate captures from Carol's archive, to whom she does not mind exposing the URI-Rs requested. Since Alice controls the MMA, she can both pre-configure the set of potential archives queried as well as provide the ability for her, Bob, or Carol to selectively aggregate from the set of archives when requesting captures for a URI-R. Were Bob uncomfortable with his aggregation requests going to Carol's archive when he used Alice's MMA, he may set up his own MMA to request captures from only his and Alice's archives without a URI-R filtering scheme like Alice's MMA. Figure 62 abstracts out the archives used for Alice and Bob's respective MMAs to conditionals.

These scenarios entail configuring a Memento aggregator with a set of archives to be queried, which is currently possible with MemGator (Section 3.1.2). However, requests sent to a MemGator instance are relayed to all archives with which the

[3]Note that MMAs do not protect the contents of an archive from being viewed, which is handled by the PWAA, to be described in Section 5.2.2.

instance is configured with every request from the client. Furthermore, the set of archives to which the request is relayed is static as was configured when initializing the service [96]. Aside from the dynamics of how a client specifies which archives to aggregate at the time of request (Section 5.1.1), Web archive users will likely not perform this sort of specification manually (e.g., specifying the Prefer header on the command-line for curl). Figure 52 shows a preliminary mockup of how a casual Web archive user may leverage this particular feature of MMAs from a Web browser. Extending on the current Mink interface that provides a mechanism for displaying memento count and navigation to view other mementos, the right side of the mockup allows the user to specify which archives are used in the aggregation process. This interface may be programmatically translated to one of the semantic and syntactic models described in Section 5.1.1 in anticipation that the endpoint (currently a running MemGator instance) understands how to interpret the archival selections.

A more scalable and decentralized approach would be to have Mink exhibit the role of an MMA. In doing so, Mink would query the archives selected and perform the aggregation in much of the same way as requesting the aggregation be performed by a local or remote MMA running outside of the browser. We are currently exploring this feature further in this proposal but anticipate it being feasible. This new capability may also enable a more user-friendly method of configuring an aggregator were running Mink instances (i.e., other users running the browser extension) able to communicate with other Mink instances running on others' browsers in a peer-to-peer manner (see Section 5.3.4).

A = Alice's archive B = Bob's archive C = Carol's archive
I = Internet Archive R = URI-R
MMA_X = Set of archives sourced for X's MMA for R
MA = Memento aggregator at `mementoweb.org`

$$\text{MMA}_{Alice} = \begin{cases} \{A, B, C\}, & \text{“facebook.com” } \in \text{R} \\ \{A, C\}, & \text{“alicesembarassingphotos.net/vacation.html” } \in \text{R} \\ \{A, B, C, I\}, & \textit{otherwise} \end{cases}$$

$$\text{MMA}_{Bob} = \{B, A\}$$

$$\text{MMA}_{Carol} = \begin{cases} \{C\}, & \text{“carolsembarassingphotos.net” } \in \text{R} \\ \{\text{MMA}_{Alice}, MA\}, & \textit{otherwise} \end{cases}$$

Fig. 62: Three Memento Meta-Aggregators are configured to perform selective aggregation.

5.2.2 PRIVATE WEB ARCHIVE ADAPTER

A Private Web Archive Adapter (PWAA) serves as the Mementity that regulates access to Web archives. Different access methods (e.g., asymmetric keys, OAuth tokenization) may be used in the implementation of authorization to a Web archive. A primary use case consists of setting up persistent access using tokenization to remove the need for reauthorization on each request. Web archives may also regulate access to a collection of private Web archives via by-design or ad hoc partitioning (e.g., collections within an archive or tagging specified URIs from a set of Web archives, respectively), producing a "key" for the subset to be used when the archive is subsequently queried. For example, for a private Web archive containing mementos for URI-Ms$_{\{1-n\}}$, a PWAA may issue a key based on the credentials supplied by the client that only allows access to URI-Ms$_{\{i,j,k\}}$ while another user assigned a different key is allowed to access URI-Ms$_{\{a,b,i\}}$. The access restrictions could also be established on a URI-R basis. The "key" concept is akin to profiles and does not require the potentially expensive procedure of subsetting to be executed repeatedly for authorization to be established. A private Web archive's primary interface is via requests from MMAs relaying requests from users.

Figure 55 shows an example CDXJ containing the `access` attributes of `type`

and `token`. These attributes for a memento specify a previously established authentication and authorization procedure with a retained token for access persistence. In this initial work, we use an OAuth 2.0 procedure to establish these attributes but the representation is extensible and not coupled to the procedure dynamics.

1. User requests captures for URI-R from MMA
2. MMA requests URI-R from Public Web Archives $Pu_{1...n}$ and Private Web Archive Pr_1
   - $Pu_{1...n}$ each return a respective set of URI-Ms $\{\{M_1\}, \{M_2\}, ...\{M_n\}\}$ to MMA
   - Pr_1 returns an HTTP 401 and an identifier for an authentication mementity (URI-P)
3. MMA returns HTTP 401, URI-P, and Pr_1 identifier to User
4. User sends credentials and URI-R to URI-P
5. Mementity at URI-P returns a token to User
6. User requests URI-R again from MMA with token and Pr_1 identifier
7. MMA requests URI-R from Pr_1 along with token
   - Pr_1 returns the set of URI-Ms $\{M_{Pr}\}$ to MMA after potentially consulting mementity at URI-P for validity
8. MMA sorts and transforms $\{\{M_1\}, \{M_2\}, ...\{M_n\}, \{M_{Pr}\}\}$ into a StarMap for URI-R
9. MMA returns StarMap to User

Fig. 63: Abstraction of the authentication to private Web archives follows a flow similar to OAuth 2.

Figure 63 describes the interaction flow of authentication and authorization to a private Web archive. This model uses the model described by OAuth 2.0 wherein the archive from which a capture is being requested takes on the roles of the resource owner and resource server (a fundamental pattern described in the specification), an

MMA or user takes on the role of the client, and a PWAA at URI-P (an identifier for an authentication mementity) takes on the role of the authorization server.

OAuth tokens as facilitated by a PWAA may be represented in StarMaps to be used in requesting URI-Ms directly from an archive after the authorization procedure has been established. In doing this, the burden of needing to repeatedly supply credentials or rely on cookies or some other state or session information for repeated access is removed from the client. Once established, we anticipate a token to be represented in StarMaps both from an MMA as well as an archive providing StarMaps directly. While the shift of burden of authorization and authentication has mostly been shifted to a PWAA from a client and archive (despite the aforementioned need to provide the token inline within StarMaps), we plan to look further into decoupling the need for amended TimeMap generation from archives to encourage adoption of the PWAA for systematic authentication access.

Adding the dimension of privacy (public/private accessibility of captures) to TimeMaps also adds another potential dimension of negotiation in Web archives beyond time. For instance, if a client desired to request only `facebook.com` with a certain time basis but only from aggregated private Web archives, the semantics do not currently exist to enable this. Beyond privacy, it may be used to consider archival content negotiation in other dimension. The mementity in the next section takes these concerns into consideration.

5.2.3 STARGATE

Memento TimeGates generally accept a URI-R and a datetime (through the Accept-Datetime HTTP header [169]) and redirect to a URI-M in return. The StarGate mementity introduced with the Mementity Framework allows negotiation in arbitrary dimensions beyond time; hence, "star" as in "*", indicating a wildcard to broaden archival negotiation beyond the temporal dimension. For public Web archives that readily return a TimeMap or a set of URI-Ms, negotiation on the dimension of time is sufficient. However, it would be both useful and necessary to perform negotiation on other additional dimensions when aggregating private and personal Web archives with captures from public Web archives. For instance, consider the scenario in Section 5.2.1 from the perspective of Alice's Web archive (and not her MMA). Alice may not want to expose the existence of URI-Ms for the URI-R `facebook.com` in her archive's holdings if a user is not authenticated to view her

archive's private captures (potentially via a PWAA). Additionally, an organization may prefer that their private archives not report even the metadata of their holdings (Section 1.3, RQ4 and RQ6), as the URI-R alone may expose the existence of sensitive information. In the above scenario, Alice was aware that she would not be returned a personalized representation when obtaining captures of `facebook.com` from IA but the exclusion of IA required explicit expression by Alice. A StarGate would allow this expression on a more dynamic basis where Alice could specify, "Only the archives that return personalized representations" instead of either, "Only my and Carol's archives" (an inclusive approach) or, "All archives for which you are configured except for Bob's and the Internet Archive", the parametric exclusion approach.

Leveraging the capability of a StarGate for negotiation beyond time has use cases beyond negotiation in the context of privacy. For instance, our previous work [98, 97] showed that nearly 85% of the URI-Ms in a TimeMap for `google.com` are redirects. For a client to have the ability to negotiate with a TimeGate to only return URI-Ms that meet a certain criteria beyond Memento-Datetime (e.g., only URI-Ms that result in an HTTP `200 OK` when dereferenced), the representation of a set of archives' holdings can be much richer in expressing metadata about the holdings. This could significantly reduce the time wasted by a user in accessing irrelevant URI-Ms (e.g., `facebook.com` login pages) and prevent misrepresentation of the quantity of captures for a URI-R [98].

5.3 MEMENTITY DYNAMICS

In previous sections we have described the fundamental functions of each mementity. In this section we will describe some anticipated dynamics of interacting with the mementities in the Mementity Framework including advanced content negotiation of Web archives (Section 5.3.1), a precedence model for advanced querying of archives for aggregation (Section 5.3.2), client-side specification of archival selection (Section 5.3.3), and collaboration and propagation of Web archives beyond the perspectives of the archives themselves (e.g., between peers, Section 5.3.4).

5.3.1 NEGOTIATION APPROACHES

Our previous work [99] discussed archival replay in dimensions like mobile versus desktop, location, etc. with emphasis on accuracy of replay, facilitated by matching

the original perspective of the capture, which is not typically exposed at replay time. Others [166, 89] have created implementations to solely interact with the capture in the original medium of the mobile Web.

In Section 2.2 we discussed the Prefer HTTP header [160], which provides a basis for content negotiation in other dimensions. In Section 5.1.1 we discussed using Prefer for client-side archival specification. Inclusion of the Prefer header requires defining preference in the Vary header of an HTTP response [160]. Though the specification consists of a registry of preferences (e.g., `return=minimal` and `return=representation`), Van de Sompel et al. [170] utilized the extensibility of the definition with `Prefer` values of `original-content`, `original-links`, and `original-headers` despite them not being registered. These `Prefer` values would hypothetically be used to obtain the raw [87], unmodified content from a Web archive instead of content that is rewritten by the archival replay system.

```
GET /starmap/cdxj/http://facebook.com HTTP/1.1
Host: stargatehost
Prefer: damage="<0.5"
Date: Tue, 04 Apr 2017 18:37:10 GMT
```

Fig. 64: A user requesting a StarMap from a StarGate where damage of all URI-Ms is less than 0.5.

Figure 64 contains a sample request made by a client to a StarGate. The request specifies that only URI-Ms with a damage score less than 0.5 are preferred. A client wishing to invoke the damage calculation procedure but limit the amount of time they are willing to wait may specify the `wait` preference [160]. In much of the same way that a TimeGate expects an `Accept-Datetime` header to perform temporal negotiation, a StarGate expects (but does not require) a `Prefer` header. Because StarGates may also perform negotiation in the dimension of time, the standard "Accept-Datetime" mechanism may be used but the additional filtering and bound specification abilities of "Prefer" are client-side specifications that we plan to investigate further with respect to the dimension of time.

```
HTTP/1.1 200 OK
Content-Type: application/cdxj+ors
Preference-Applied: damage="<0.5"
```

Fig. 65: Upon completion of the potentially temporally expensive procedure of calculating damage for all URI-Ms for `http://facebook.com` from Figure 64, a StarGate will respond with headers containing the applied preference.

For computationally expensive processes like damage calculation for a large set of URI-Ms, a StarGate may immediately respond with an HTTP status `202 Accepted` to indicate that the request has been accepted for processing but the processing is not yet complete. Subsequent accesses using the same request in Figure 64 prior to the StarGate's completion may return a `102 - Processing` status [70]. When a preference has been applied to a requested StarMap from a client to a StarGate, the response will contain the `Preference-Applied` HTTP response header [160] and an HTTP 200 (Figure 65). We anticipate this preference being propagated to the list of metadata headers in a CDXJ StarMap (similar to those highlight in red in Figure 57). We are currently investigating a scalable, extensible, and semantic way to accomplish this while adhering to the CDXJ syntax.

5.3.2 PRECEDENCE MODEL

Private Web archives contain an inherent characteristic where exposing the metadata about an archive's contents could be sufficient to identify the archive's contents. For example, a private archive responding with a StarMap containing URI-Ms for captures of my online bank statement would reveal that I am preserving personal banking information (or, with fewer ramifications but still a need for privacy, a site with embarrassing photos).

A second aspect exists independent of exposing the metadata that may reveal a private Web archive's contents. Were a client to setup a Memento aggregator inclusive of their private Web archive, they may prefer a mechanism that returns the results only from their private archive if it contains contents for a given URI-R and only default to sending the request to public Web archives if no results were returned. The set of archives queried may have a tiered request configuration with requests being performed in a more synchronous procedure with the aforementioned

short-circuiting procedure applied.

Figure 66 illustrates requests being first sent to the private archives then to public Web archives. It may also be desirable to allow this behavior to functionally coexist with conventional pipelined asynchronous archive querying. As with the Snowden Archive-in-a-Box [115] example in Section 2, access to this content if checking for the existence for captures in other archives may imply interest or association with the subject matter, in some cases itself being revealing or even incriminating.

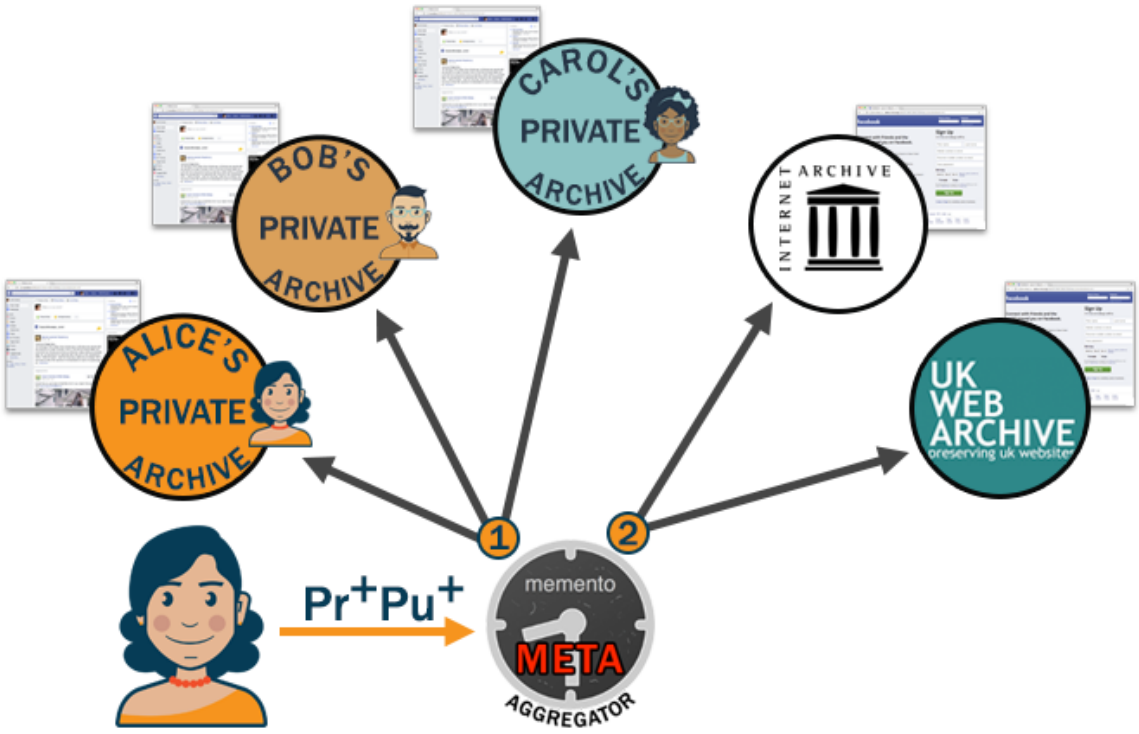


Fig. 66: Archival precedence using private first then public Web archiving querying model ($Pr^{+}Pu^{+}$).

For a Memento aggregator to include contents or simply metadata from the Snowden archive along with other personal, private, and public captures would require special handling to be considered when accessing resource from the Snowden archive. For example, a user may want requests for a certain set of URI-Rs to not also be requested from other private Web archives beyond the Snowden archive or their own personal Web archive for the sake of privacy of the request.

We propose two initial approaches to accomplish this: explicit specification by a client at the time of request and analysis of mementos with a potentially personalized representation. For the latter, we [99] identified three methods for identifying

personalized representations. Of the methods proposed, one not investigated further (we opted for one of the other three) was to be able to specify additional environment variables when selecting a representation of a resource. The downside, we mentioned, was the requirement of a specialized client. The specialized "client" in this case may be the mementity responsible for determining the degree of personalization of the representation, i.e., the StarGate.

When aggregating and replaying a URI-R over time from a set of archives consisting of captures from both public and private Web archives, it may be desirable to first check for private captures prior to requesting URI-Ms from public Web archives (Figure 66). For example, in aggregating URI-Ms for `facebook.com` that include mementos of my news feed from my private archive and unauthenticated login pages from institutional public Web archives (Figure 10), the latter is less useful in observing how the page has changed over time. To maintain relevancy of the desired sort of representation, we may want to check for the existence of captures from private Web archives *first* and then, only if none are present, resort to requesting the captures consisting of a login page. This model of precedence (request priority) and short-circuiting (stop requesting captures if a condition is met) via Memento aggregators does not currently exist but could be critical in a user expressing what they expect from an aggregator beyond simply mementos for a URI-R.

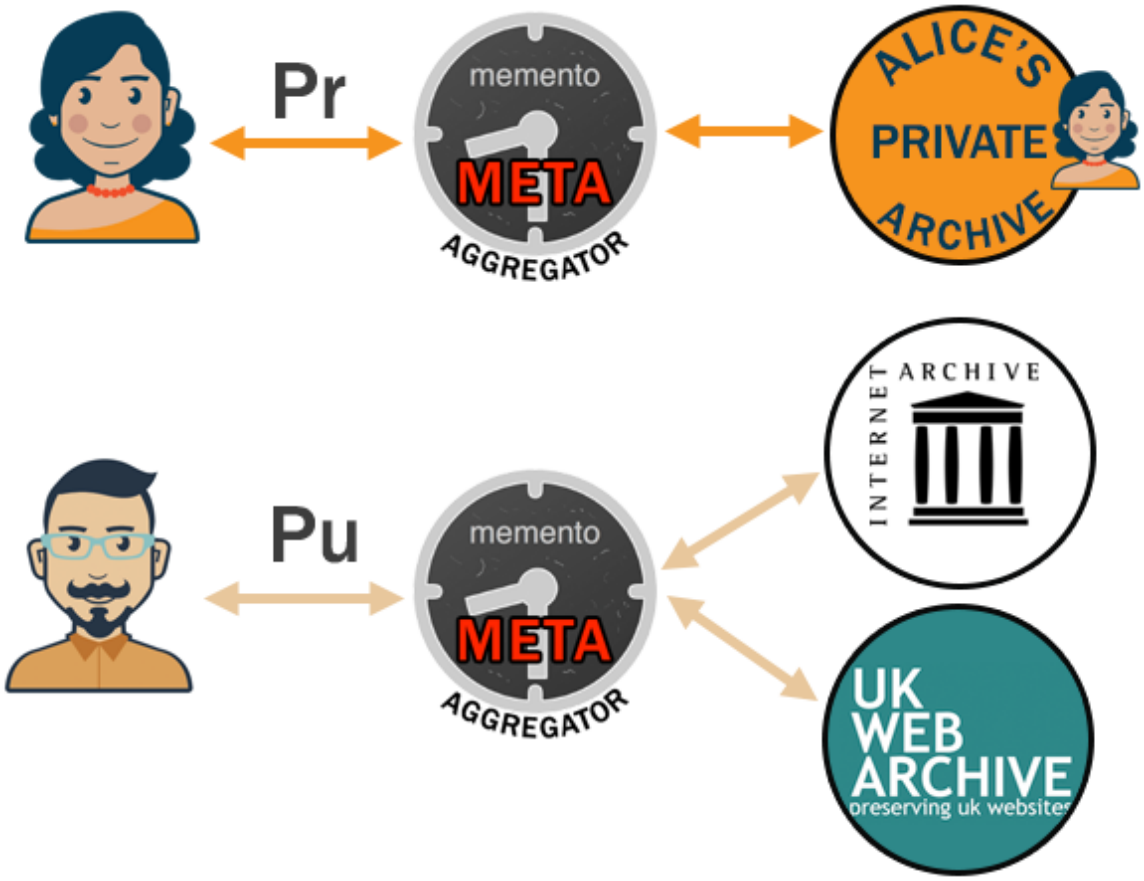


Fig. 67: PrivateOnly and PublicOnly aggregation in an MMA.

In the basic model below, we express various access precedence models (henceforth *profile*) for containing boolean categorization of private and public Web archives. In each profile, order is significant and thus a simple regular expression can be used where P_u symbolizes a public Web archive endpoint, P_r a private Web archive endpoint, and the "+" superscript indicating at least one or more consecutive instances.

$$noArchives \rightarrow \varnothing \rightarrow \{\} \tag{7}$$

$$publicOnly \rightarrow {P_u}^+ \tag{8}$$

$$privateOnly \rightarrow {P_r}^+ \tag{9}$$

$$privateFirst \rightarrow {P_r}^+ {P_u}^+ \tag{10}$$

$$publicFirst \rightarrow {P_u}^+ {P_r}^+ \tag{11}$$

The basic profiles pair with the syntax of the `profile` relation type [174], allowing clients to request resulting TimeMaps containing URI-Ms from a subset of archives from which the Memento mementity requests (Figure 67). The preliminary scheme for short-circuiting of subsequent requests is also boolean, e.g., requests should only be made to public Web archives when the `privateFirst` profile (Equation 10) is specified by the client when no identifiers for captures are returned from private archives. This model also assumes that the sets P_u and P_r are disjoint ($P_u \cap P_r = \varnothing$) for simplicity, but this may not be the case in reality. For Web archives that contain both private and public captures, an approach toward achieving mutually exclusivity could be to separate each set of the private and public URI-Rs into an abstraction of separate collections. For example, as discussed earlier, the UK Web Archive contains captures from its legal deposit with restricted off-site access; that is, a user cannot access the mementos unless physically on location at the library (Figure 28).

5.3.3 MMA ARCHIVE SELECTION

Here we revisit the scenario introduced in Section 5.2.1, and abstracted in Figure 62 to show how an MMA can perform selective aggregation. Alice sets up an MMA (MMA_{Alice}) that is configured to request captures from her archive (A), Bob's archive (B), Carol's archive (C), and the Internet Archive (I). For some URI-Rs, like `facebook.com`, it may not make sense to aggregate Alice, Bob, and Carol's captures with those from Internet Archive, so she can specify a rule of only aggregating mementos from {A, B, C} when those URI-Rs are requested. For other URI-Rs, like `alicesembarrasingphotos.net`, Alice may want to prevent exposing the fact that she is looking for certain old captures to Bob and the Internet Archive, but wants to also aggregate captures from Carol's archive, with whom she does not mind exposing the URI-Rs requested. She does this by creating another rule to only aggregate from archives {A,C} in those cases. By Alice controlling the MMA, she

can both pre-configure the set of potential archives queried as well as provide the ability for her, Bob, or Carol to selectively aggregate from the set of archives when requesting captures for a URI-R. Were Bob uncomfortable with his aggregation requests going to Carol's archive when he used Alice's MMA, he may set up his own MMA (MMA_{Bob}) to request captures from only his and Alice's archives without a URI-R filtering scheme like Alice's MMA. Carol also sets up an MMA (MMA_{Carol}) that defaults to using Alice's MMA and the `mementoweb.org` MA except when requesting URI-Rs from `carolsembarrassingphotos.net`.

As an endpoint, MMAs may aggregate and request access to captures to private Web archives using a token-based authorization model (e.g., using OAuth as described in Section 5.2.2). The query may be subsequently routed to an applicable and corresponding Web archive (private or public) after authentication has been established. MMAs may query other MMAs with the expectation that the results returned will be consistent with those from an MA with additional indicators for content beyond the scope of an MA (e.g., a flag for content from a non-aggregated or public archive). In the scenario above, Carol may want additional archives aggregated beyond the default case in Figure 62 so she can utilize the ruleset of Alice's MMA, as well as add filtering rules of her own. The filtering that an MMA performs may not be (and more likely is not) exposed to clients or other MMAs that look to it as a source for URI-Ms. Doing so would be a detriment to the function of an MMA preventing selective aggregation, though does not prevent clients from accessing the aggregated archives directly. Note that in the case of Carol's MMA, there exists a redundancy in that both Alice's MMA and the `mementoweb.org` MA will request URI-Ms from IA. While Carol's MMA may perform an operation to consolidate duplicates (i.e., a "UNIQUE" operation), time may still be wasted waiting for all archived sources to respond to requests to Carol's MMA. Carol may also only want to look to some archives if none, too few, or some other quantifier or qualifier exists in an initial set or series of archives. A StarGate may be used for advanced querying of this sort.

5.3.4 COLLABORATION AND PROPAGATION

Collaboration in Web archives is often exhibited by individuals and organizations submitting URIs to a service to preserve, particularly when a significant event is anticipated or occurring. In addition to providing novel approaches (beyond simply

submitting URIs) for collaboration by-reference, in this proposal we plan to focus on collaboration of Web archives by-value. In Section 4.3 we introduced InterPlanetary Wayback for propagation of personal Web archives. This propagation may be accomplished by-reference where the reference identifier consists of a content addressed hash uniquely identifying the archived content. By utilizing the mementities in Section 5.2, particularly the Memento Meta-Aggregator from Section 5.2.1, a user may tailor the StarMap advertised to provide implicit guidance for those wishing to locally copy and further disseminate a personal Web archives' mementos. This propagation of a capture exhibits a form of collaboration through continued accessibility of mementos in personal Web archives.

The crux of ipwb is for decentralizing and distributing mementos that reside in accessible WARCs (Section 4.3). Our initial approach at privacy in ipwb entailed performing symmetric encryption to the content prior to disseminating it into IPFS [101]. Using this method allows Alice to share her ipwb CDXJ with Carol for Carol to "pull" the captures for local propagation from Alice's machine via IPFS. Figure 68 shows Alice pushing her local WARCs containing her private Facebook captures to ipwb using encryption by setting a flag upon ipwb invocation. Alice is returned a CDXJ, which she can then transfer to Carol. Upon receipt, Carol can instruct her local ipwb instance to replay the CDXJ. Carol may attempt to access the mementos described in her CDXJ, whose header and payloads are retrieved from IPFS via ipwb but still encrypted. Carol must know the encryption key to be able to interpret the payload, whose decryption and transformation is handled by the ipwb replay system.

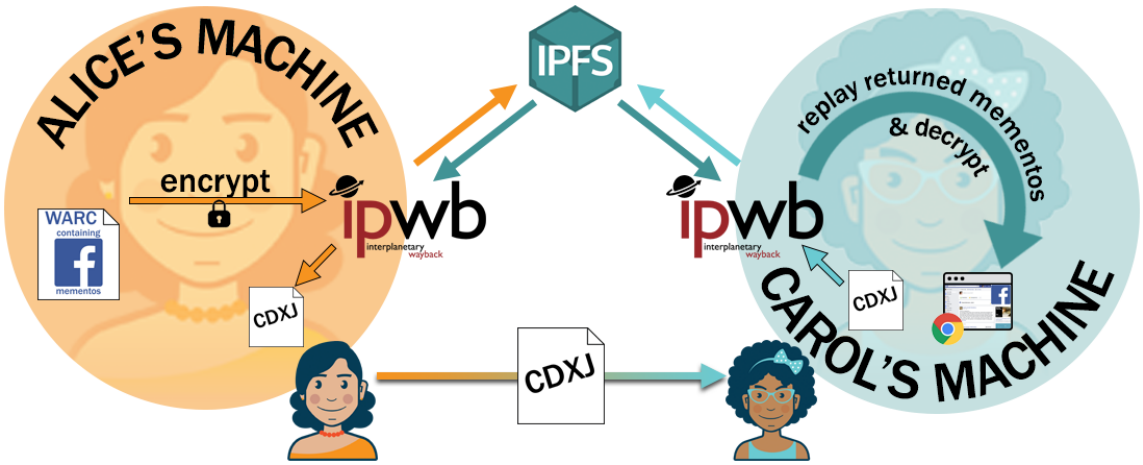


Fig. 68: The extended ipwb model for collaboration involves symmetric encryption and decryption of the payload prior to dissemination. When Alice transfers the CDXJ generated from pushing her Facebook WARCs to IPFS via ipwb (specifying the encryption flag), she may then transfer the CDXJ to Carol. Carol can then decrypt the payload when replaying the mementos described in the CDXJ.

In this proposal we hope to make this process more seamless by obscuring the details of CDXJ propagation (which may be unnecessarily technical for users) to emphasize the collaboration process. We plan to explore a mechanism for discovery of other ipwb instances for collaboration of the aggregate distributed picture of the archived Web using ipwb. Doing so will require a decentralized approach to maintain the ethos of IPFS and distributed personal archives instead of relying on a centralized server for cataloging instances. This exploration may also reveal other opportunities for personal archive discovery outside of the realm of ipwb, which could be a significant contribution to personal Web archiving.

In Section 5.3.3 we briefly discussed the capability of Mink instances exhibiting the capabilities of a MMAs communicating with one another for peer-to-peer, purely client-driven archival querying and aggregation. In seemingly unrelated previous work, we leveraged a then-young WebRTC protocol to facilitate the replication of NASA satellite imagery posted to the Web [93]. In addition, we [7] have explored using Web and Service Workers in the context of Web archives and applied this functionality for client-side processing of mementos – in this case resolving absolute URIs to be rerouted (instead of rewritten) to the local replay system.

An advancement in IPFS since the creation of ipwb is the use of Service Workers for client-to-client communication using WebRTC in the JavaScript implementation

of IPFS [139]. We plan to investigate this capability through implementation prototyping to extend Mink to leverage this sort of communication while additionally interfacing with the peer-to-peer personal archive discovery and propagation in ipwb as described above. The browser extension medium may be more accessible for casual users and may facilitate more users collaborating and propagating their captures compared to ipwb, which requires a local installation outside of the Web browser.

5.4 USER ACCESS PATTERNS

This section describes various User Access Patterns for Web archives, some currently in-practice and others anticipated and facilitated with the implementation of the Mementity Framework we describe in this research. Figure 69 shows a composite hierarchy that illustrates how each of the patterns may relate when applied. The patterns to be described are:

Pattern 1: Single archive access (Section 5.4.1)

Pattern 2: Aggregation of multiple Web archives (Section 5.4.2)

Pattern 3: Aggregator chaining (Section 5.4.3)

Pattern 4: Aggregation with authentication (Section 5.4.4)

Pattern 5: Aggregation including a hybrid public-private archive (Section 5.4.5)

Pattern 6: Aggregation with filtering via MMA interaction (Section 5.4.6)

Pattern 7: Aggregation with filtering via SG interaction (Section 5.4.7)

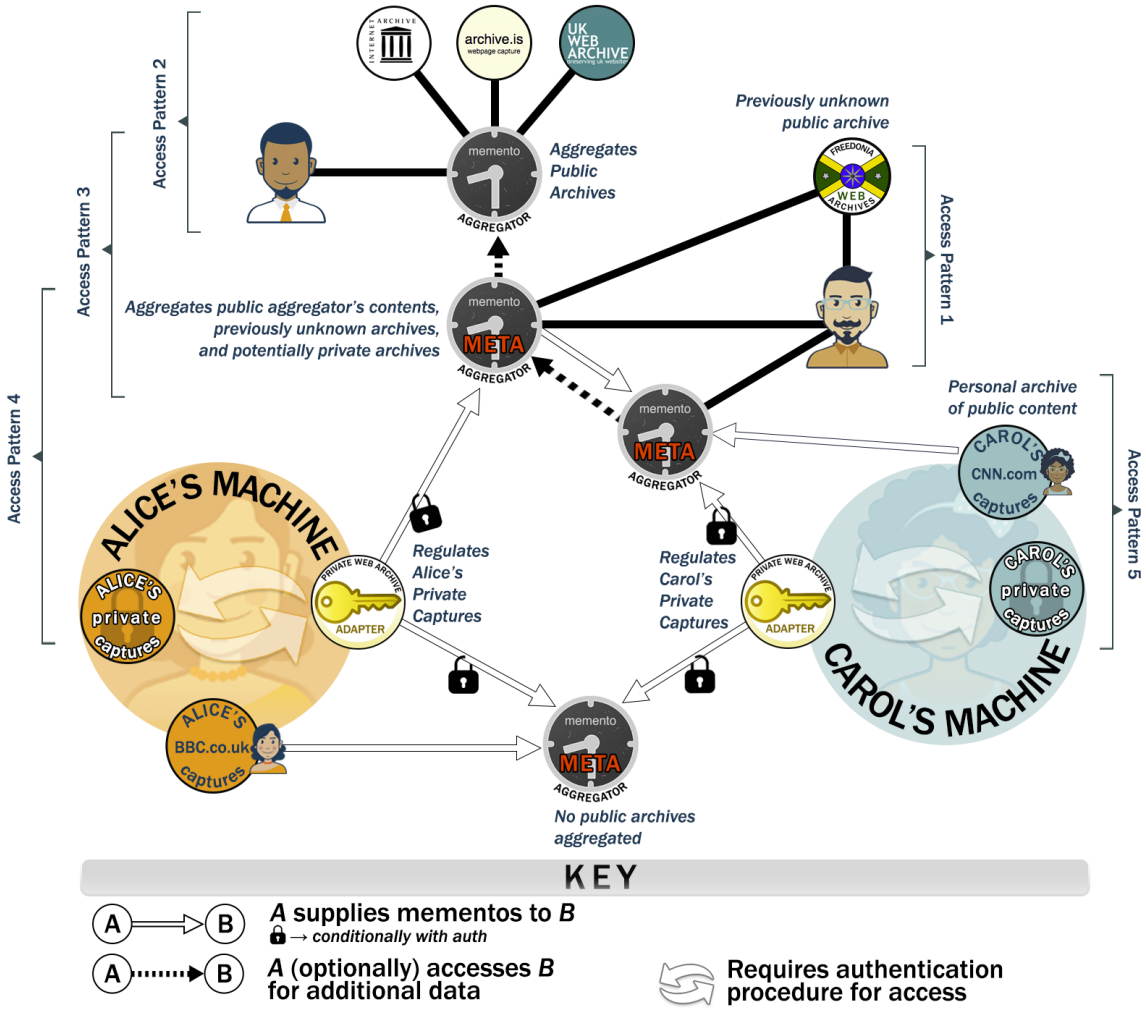


Fig. 69: MMAs and PWAAs form a hierarchy of access for a variety of scopes of Web archives. User Access Patterns from Section 5.4 are shown to regulate access to private Web archives for aggregation with public Web archives without changing the functionality of the infrastructure in-place (e.g., Wayback deployments, Memento aggregators, etc.).

5.4.1 PATTERN 1: SINGLE ARCHIVE ACCESS

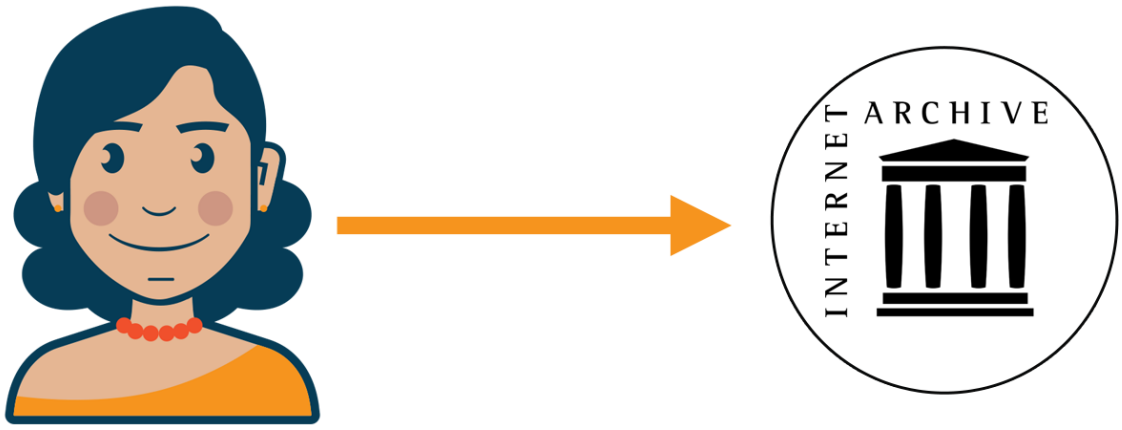
Conventional direct access by a user to a Web archive (e.g., Internet Archive) defines an initial familiar existing User Access Pattern. In this scenario, a user performs an HTTP request for a URI-M from a Web archive using the user agent of their choice (e.g., curl, Google Chrome) and is returned a memento. For example, to obtain one of the captures for `nasa.gov` shown in Figure 6, we sent a request

to a URI-M[4] for the URI-R nasa.gov. Figure 70b shows the archived representation, a familiar Web page representation, that is returned to a user when accessing this particular URI-M. Figure 70a shows this symbolically through a user accessing an archive. The symbolic representation in this figure is a fundamental base case that will be built upon in this section. Access to individual, publicly available Web archives inherently requires no aggregation of multiple Web archives (and thus, no aggregator mementity). Pattern 1 serves as a basis for further patterns. This pattern is intentionally generic in that it accounts for access by a user to both institutional and personal Web archive instances. The pattern also conceptually encompasses access from a number of endpoints, e.g., an archive's Web interface, via selection of a URI-M from a TimeMap, etc. Finally, this pattern is not limited to accessing public or institutional Web archives. For instance, a WAIL (Section 4.1.3) user may preserve a Web page of their choice and access the memento from the replay system accessible at http://localhost on their own machine.

5.4.2 PATTERN 2: AGGREGATION OF MULTIPLE WEB ARCHIVES

Memento facilitates the aggregation of identifiers for captures from multiple Web archives. Memento aggregation is accomplished through combining URI-Ms as well as other metadata from multiple Web archives' Memento TimeMaps. For example, requesting an aggregated TimeMap of the URI-R matkelly.com from a public Memento aggregator may return the TimeMap shown in Figure 24. Note the URI-Ms listed are from a variety of public Web archives. Memento proxies [1] also exist to adapt the responses from Web archives that have not yet implemented Memento. Figure 71 shows a user accessing a public Memento aggregator, which aggregates captures from three public Web archives (IA, UKWA, and archive.is). Beyond Patterns 1 and 2 resides the contribution of the Mementity Framework.

[4] http://web.archive.org/web/19981202170636/http://www.nasa.gov/



(a) A user (with an implied user-agent) accesses an archive directly.

(b) Accessing `nasa.gov` as it appeared on December 2, 1998 using Google Chrome.

Fig. 70: Access Pattern 1 (Section 5.4.1) describes current fundamental access of a memento. A user often experiences this through a Web browser (b) but other means (e.g., curl) represent the same access pattern (Section 2.2).

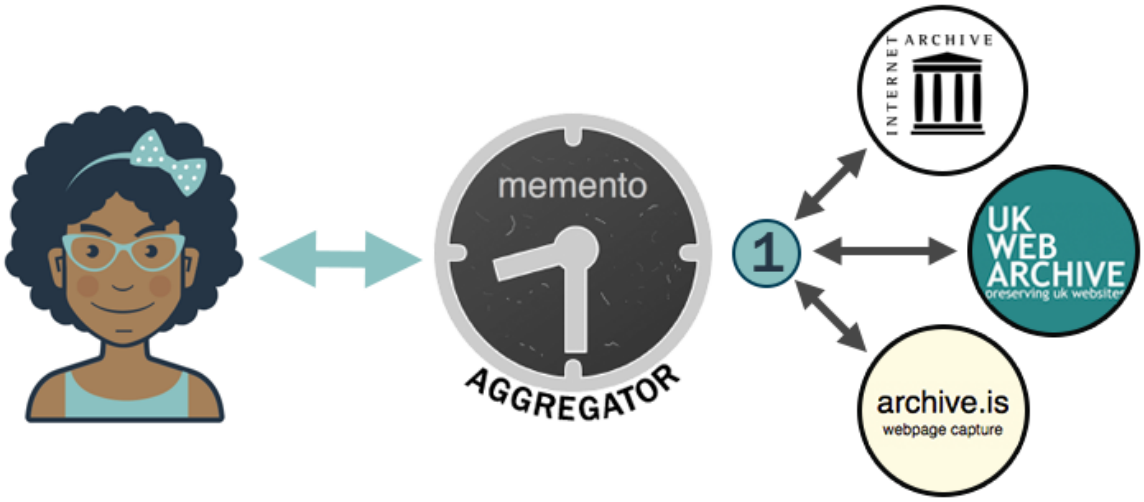


Fig. 71: Access Pattern 2 (Section 5.4.2) represents a user accessing a Memento aggregator to obtain aggregated results from a set of archives. The archives contained in this set are often not customizable by the user. A TimeMap is returned to the user containing URI-Ms and other Memento metadata (e.g., original URI-R). A user may then access a URI-M contained in the returned TimeMap (Section 5.4.1). This pattern exhibits an equally-weighted querying model without precedence (requests are executed in parallel) or short-circuiting.

5.4.3 PATTERN 3: AGGREGATOR CHAINING

A user may initially access an MMA instead of an MA per Pattern 2. Figure 72 pictorially describes an MMA relaying a request for URI-Ms for a URI-R from a user to the aforementioned MA. The MA performs the query and returns the results to the MMA. The MMA then relays the results to the user. This pattern introduces simple hierarchical chaining of aggregators and is novel to the introduction of an MMA. In the scenario described in Section 5.2.1, a use case for aggregator chaining without supplementing the results would be the exclusion of certain archives from the results. If Carol sets up her MMA to request captures only from the `mementoweb.org` MA, but at request time specifies that she wants to exclude all results from `archive.is`, she may do so using this chaining Pattern.

Per Section 5.2.1, a MMA is a functional superset of an MA. Because of this, the MA in Figure 72 could be replaced with an MMA, configured to request captures from the same Web archives, and retain the same dynamics initially described above for this pattern.

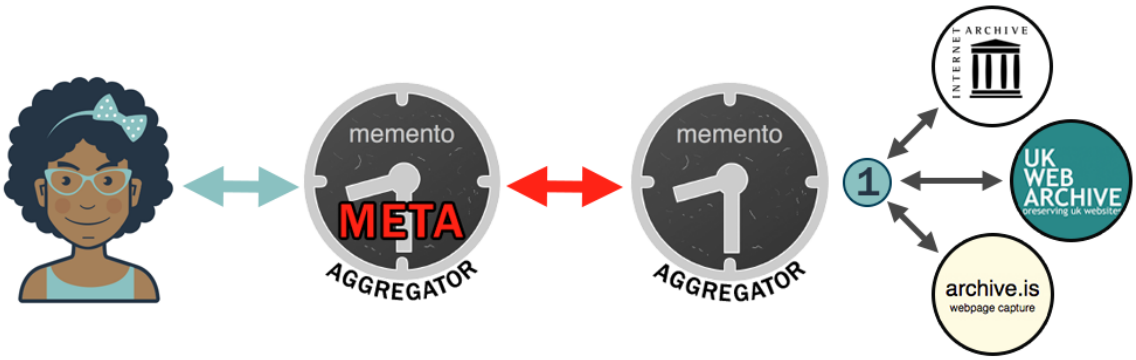


Fig. 72: A Memento Meta-Aggregator (MMA) acts as a functional superset for a conventional Memento Aggregator (MA). This attribute allows an MMA to replace an MA with extended features beyond the scope of a conventional MA. An MMA can also acts as a simple relay of the results (pictured) with the potential for a user to modify the set of Web archives aggregated at a later date – a function not available for MAs that a user does not control.

Aggregator chaining also opens the potential for supplementing of results. MMAs allow for runtime inclusion of additional Web archives for aggregation through specification by the user. This feature may not be scalable (which we are investigating in this proposal) but is intended for personal deployment of MMAs and is mitigated by caching [49] and deeper levels of MMA chaining. Consider again the scenario where Carol wished to exclude the `archive.is` captures. She may also configure her aggregator (an MMA) to request captures from her archive to be aggregated with the captures from the `mementoweb.org` aggregator minus the `archive.is` captures, as expressed in the request to the aggregator. Figure 73 shows a scenario where an MMA is configured with the inclusion of an additional Memento compatible public Web archive, "Freedonia Web Archives", with which the MA is either not aware or does not aggregate by default. Along with relaying the request from the user for mementos for a URI-R to the MA, the same request is sent to the Freedonia Web Archive from the more inclusive MMA in the hierarchy. Upon obtaining a response from both the MA and the Freedonia Web Archives for URI-Ms for a URI-R, the MMA aggregates these results and returns them to the user.

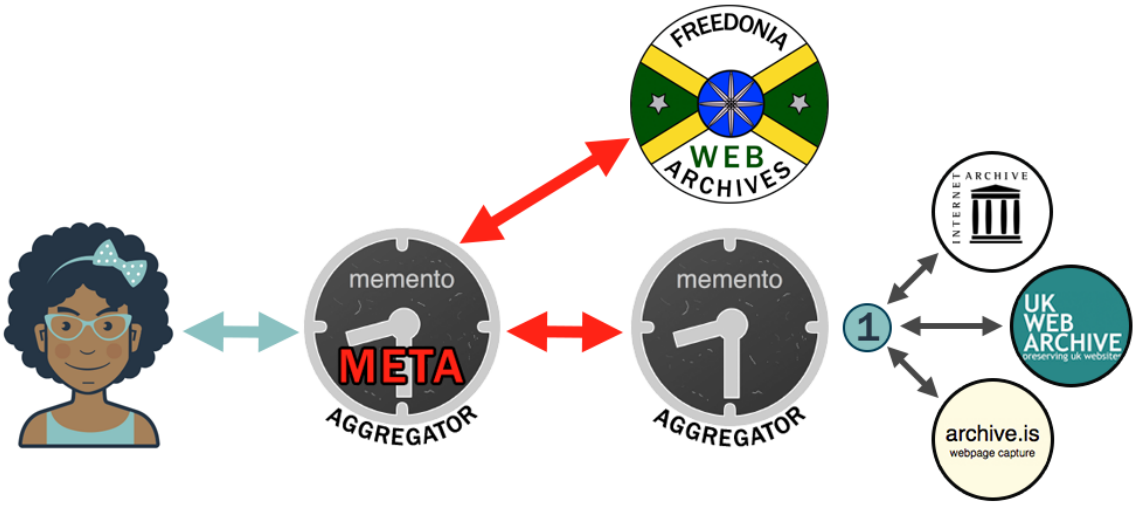


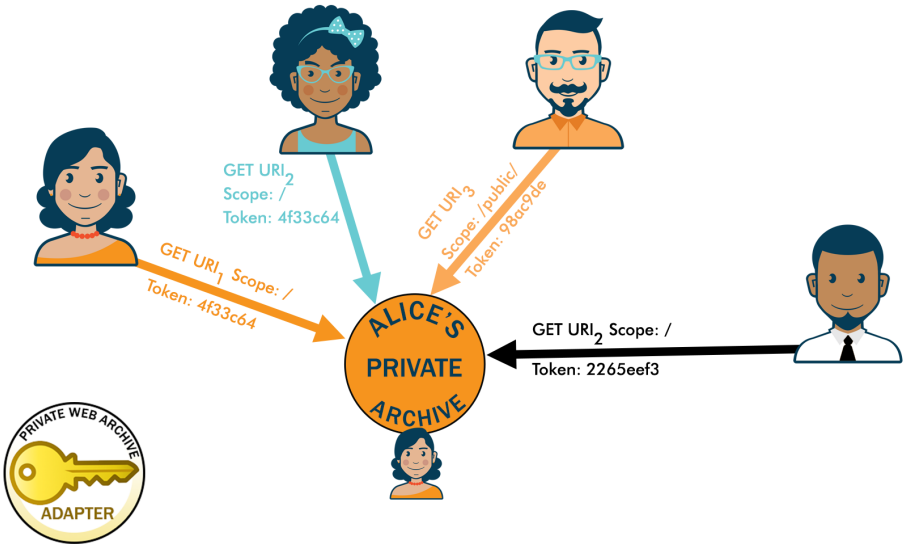
Fig. 73: Chaining Memento Meta-Aggregators allows results to be supplemented. Using a hierarchical MMA approach, a previously unaggregated public Web archive may be aggregated with the results for a URI-R from a conventional Memento aggregator. Pattern 4 extends on this base relationship between MMAs (shown in Figure 72) by an MMA adding the URI-Ms and other Memento metadata from a new previously unaggregated (the fictitious yet publicly accessible) Web archive into its results. Accessing the MMA in this figure would yield results from four archives whereas a user requesting an aggregated TimeMap from the MA would contain results from only three archives.

5.4.4 PATTERN 4: AGGREGATION WITH AUTHENTICATION

All previous patterns in this section have been described with the assumption that an archive will return a TimeMap of its captures for a URI-R or a URI-M if supplied a URI-R and datetime. As previously discussed (e.g., scenarios in Section 1.2, enabling the personal Web archivist in Section 4.1), aggregating or simply accessing private Web archives (the latter per Pattern 1) requires systematic regulation to ensure the potential privacy features of the captures are being considered. This pattern describes a potential method for answering RQ5.

To extend the idea to aggregation with authentication, it is useful to first consider the scenario where Bob attempts to access the mementos of Alice's private Web archives. Alice has configured her archive to integrate with a PWAA (Section 5.2.2). Because of this, the fundamental Access Pattern 1 does not apply. Bob will experience the authentication flow described in Figure 56 and be required to supply credentials to access the captures. Upon successful authentication, he will be issued

a token, which may be reused for future access until either expired or revoked. In the situation where Alice has accessed her own archive and received a token, she may share this token for access with Carol (Figure 74). Upon the archive receiving a request from Alice or Bob, who have separately authenticated, or Carol, who is reusing a token (Figure 74), the archive will consult its configured PWAA to validate the token and the scope of the request. This pattern will apply to any other users attempting to access the archive (e.g., Malcolm in Figure 74), who may be rejected access if a token is not supplied or an invalid token is supplied (as configured).

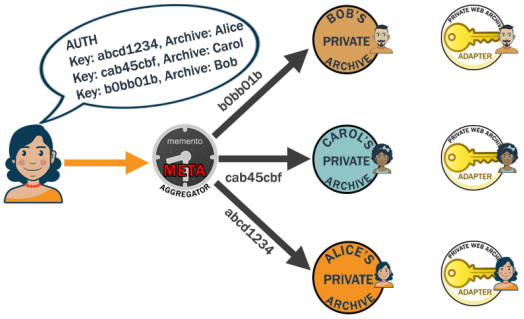


(a) Four users (left to right: Alice, Carol, Bob, Malcolm) request URIs in Alice's private archive using pre-established tokens and a defined "scope". Scope here is reused from the OAuth specification to potentially limit access to parts of an archive on a token basis.

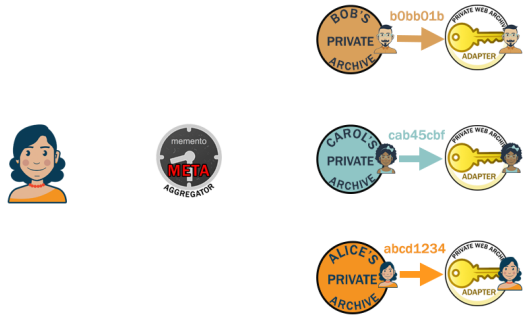
(b) Prior to authorizing access, a private archive will consult its PWAA to verify access to the scope and URI using the respective token suppplied.

Fig. 74: A token obtained from the process in Figure 56 can be shared and reused for persistent access. Accessing the PWAA responsible for access control of a private Web archive will initially deny access without providing credentials. Tokens may be revoked and re-established, allowing regulation of access to private archives. Requests shown as temporally parallel for graphical simplicity but more likely performed at different times.

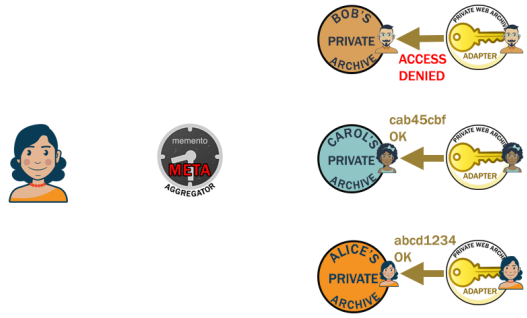
The above scenario is the core of the Pattern where an MMA aggregates captures inclusive of one or more private Web archives configured as described. Extending the aggregation with authentication pattern to access to multiple private Web archives from a single user is shown in Figure 75. Here, Alice has pre-established authentication with her own private Web archive, Carol's private Web archive, and Bob's private Web archive with keys/tokens of `abcd1234`, `cab45cbf`, and `b0bb01b`, respectively. She has configured her MMA to *only* access these three archives for results of queries for mementos. She supplies these keys to her MMA (Figure 75a) at the time of request, which are relayed to each respective archive per the role of the MMA. The private Web archives each consult their respective PWAA to validate the key that Alice supplied (Figure 75b). Both Alice and Carol's PWAA validate their respective keys but Bob's PWAA rejects the key supplied by Alice (Figure 75c). With the token and thus the request validated, Alice and Carol's private Web archives supply StarMaps with 10 and 3 mementos to Alice's MMA (Figure 75d). Bob's archive, having received the instruction to reject the authentication supplied with the request, returns either no response, a response with 0 mementos, or an unauthorized response per the implementation of his archive. In some scenarios, one sort of these responses may be preferable to another, for instance, when not wanting to disclose the reason for response rejection.



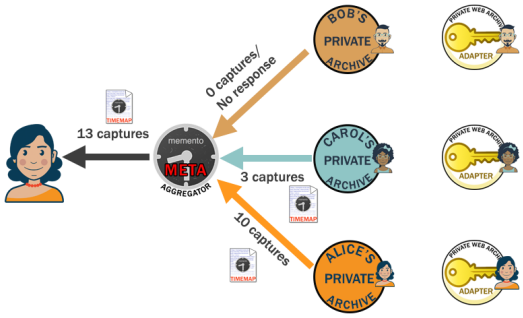
(a) A user provides different keys for three different private Web archives, aggregated by an MMA



(b) The private Web archives check the validity of the tokens/keys submitted with their respective PWAA.



(c) Two archives validate the keys while a third has expired, been revoked, or is invalid.



(d) When the three private Web archives' contents are aggregated, no results are returned from the archive whose token did not validate

Fig. 75: In instances where an MMA is configured to only aggregate private Web archives or the *privateOnly* short-circuiting (Section 5.3.2) directive is supplied, a user may specify different keys on a per-archive basis.

5.4.5 PATTERN 5: AGGREGATION INCLUDING A HYBRID PUBLIC-PRIVATE ARCHIVE

Pattern 5 exhibits situations where a user queries a Memento Meta-Aggregator with no or insufficient credentials but still retains access to publicly exposed content in a private Web archive. Consider a scenario where Alice has a single archive consisting of some mementos she does not mind being available (e.g., her `cnn.com` captures) and some she would rather not be shared (e.g., her `facebook.com` captures). These captures may be separated into a collection or separate "sub-archives" within her archive but more likely these captures are intermingled. In any of these

cases (collection-based, sub-archives, or intermingled), regardless of how captures are organized, a user may want to determine the accessibility of the various ad hoc sets of captures.

The need for finer grained control of access beyond URI matching (e.g., all URI-Ms for `facebook.com` have restricted access) may be more apparent with an example where live Web access control is not carried over to the archived Web. Carol is preserving her `youtube.com` channel inclusive of her publicly accessible videos, private videos (only available to select users on the live Web), and unlisted videos (videos not indexed but accessible to anyone with the video page's URI-R). All three classes of videos would be accessed on the live Web at a URI-R pattern similar to `https://www.youtube.com/watch?v=cYZSx5TyL1k` where the value of the `v` query string parameter is representative of a unique identifier for the video on `youtube.com`. However, without Carol whitelisting a user (e.g., Alice may access the video using her own YouTube account) or being a user herself (with implicit access as the author), the private video would not be accessible on the live Web. In the scenario where this video is archived by Alice or Carol, the live Web access restriction would not be retained and the private video would become accessible on the archived Web without access restrictions in-place (Figure 76). The "unlisted" concept on `youtube.com` also introduces a dimension of necessary restriction beyond simply public and private – a user must know the URI-R of the unlisted video to access it. The distinction beyond the live and archived Web URI-R and URI-M (respectively) is moot here, however, for the URI-M to be listed in a StarMap would indicate it existed (similar to the `alicesembarassingphotos.net/vacation.html` scenario in Section 5.2.1). Further, the access restrictions on the archived Web need not follow the degree of accessibility of the videos on the live Web. For example, a user may want a privately archived unlisted video capture to not be accessible despite knowing the URI-R on the live Web being sufficient for access.

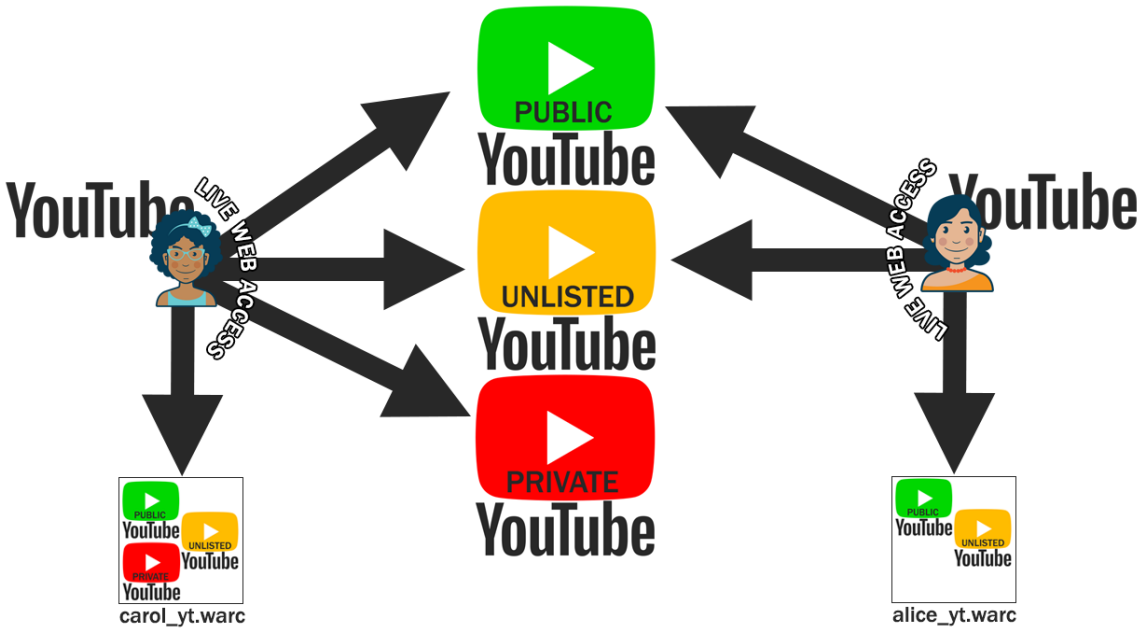


Fig. 76: A user may want finer grained access control of captures within her archive without the need for separate collections. Carol has preserved her public, private, and unlisted `youtube.com` videos but may wish to restrict each class's accessibility within their archive. Alice can access the public and unlisted videos but the question remains as to whether the unlisted video should be publicly available on the archived Web.

MMAs may organically induce this pattern when aggregating captures from multiple archives with a mix of access scenarios. Extending on the above scenario, Alice archives her personal `youtube.com` channel as well but has only private videos that she shared on the live Web on a basis of her choice. Alice has setup an MMA to aggregate her archive's captures. She is content with some of the videos being publicly accessible from her archive (e.g., public videos, green icons in Figure 76) and some being selectively accessible to those who authenticate with her archive's PWAA (e.g., yellow and red icons in Figure 76), despite all videos being private on the live Web. Carol, Alice's sister, wishes to utilize Alice's MMA to aggregate some of Alice's private family videos with her own private family videos. Carol may also want to utilize Alice's MMA, as Alice may have have configured her MMA to be more permissive of requests from her own MMA compared to outside requests from others' MMAs or other individual users.

For simply acquiring the relevant captures from Alice's MMA, Carol would follow the model in Figure 63. However, to aggregate her own and Alice's captures, Carol

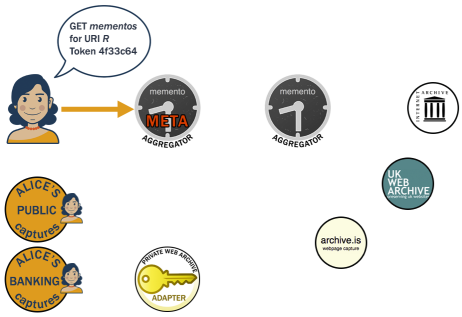
would configure her own MMA to request captures from her archive as well as Alice's MMA. The Figure 63 procedure may need to be repeated for each archive aggregated to establish persistent and secure access but this pattern would allow Carol to accomplish the sort of aggregation of private captures (even if they were public on the live Web) she desires.

Each archive in the set of archives aggregated by an MMA may require special handling, as exhibited by the aforementioned YouTube scenarios. Figure 77 depicts Alice accessing an MMA that in turn retrieves captures for an MA and Alice's own captures. Alice's captures here, despite being in the same "archive", are distinguished between her private (e.g., banking) captures and her public (e.g., CNN) captures. At the time of request, Alice supplies a token to the MMA (Figure 77a), which is only relayed to the corresponding archive (Figure 77b) and not propagated to where it is inapplicable. While Alice's archive may asynchronously return three results (Figure 77c) for the request while the request is still being propagated to the other archives via the relay to the MA, Alice's private Web archive may begin authenticating the token Alice supplied (per Figures 56 and 74). Upon successful authentication by the PWAA (Figure 77d), the private captures may be returned to the MMA. In the same step, the MA will have received results from three archives with 100, 30, and 10 mementos. Figure 77e depicts both Alice's private captures (10,000 in number, given the content is personal and Alice is diligent about archiving) being returned to the MMA as well as the MA aggregating and returning the TimeMap with 140 captures from the public archives. Finally, the results from Alice's archives (10,003 mementos) and the public archives (143 mementos) are aggregated (Figure 77f) and returned to Alice in a StarMap containing 10,143 URI-Ms.

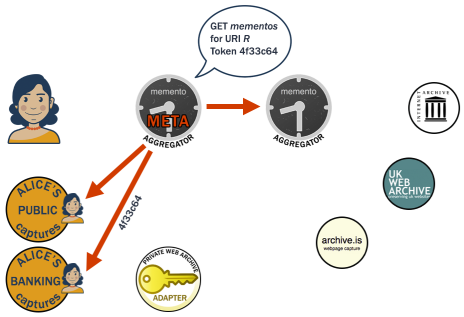
5.4.6 PATTERN 6: AGGREGATION WITH FILTERING VIA MMA INTERACTION

In Section 5.3.1 we briefly discussed negotiation approaches with regards to additional dimensions to be represented in StarMaps. In this pattern we will describe negotiation in the dimensions represented by each of the attribute types in Section 5.1.2.

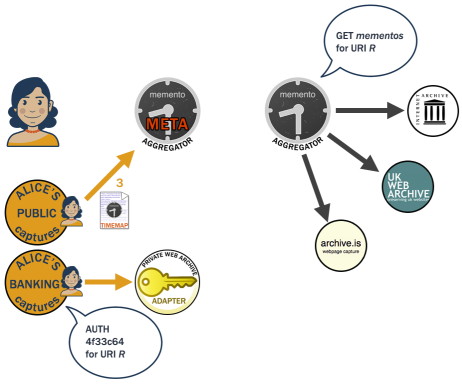
Pattern 6a involves pre-filtering URI-Ms based on access attributes. In this subpattern, Alice sends a request for a URI-R to an MMA with the HTTP request header `Prefer: privateOnly`. The MMA send a request to an SG with the archives it



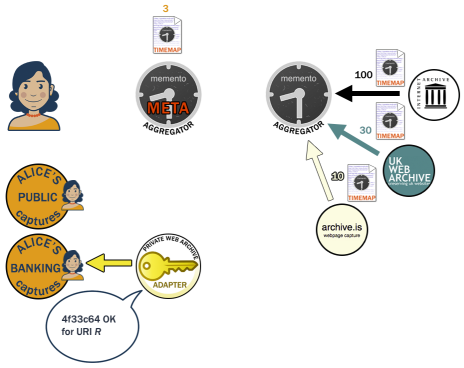
(a) User supplies a URI-R and a pre-obtained token after performing the procedure in Figure 56.



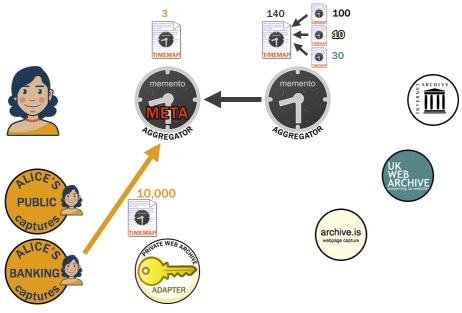
(b) The URI-R and token are relayed (where applicable) from the MMA to the mementities (two archives and an MA).



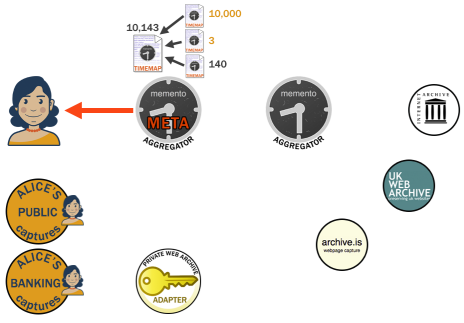
(c) MA requests URI-R from archives, one archive returns results directly to MMA, a private archive verifies the token with an associated PWAA.



(d) Archives return results for URI-R to MA, PWAA confirm token thus authorizing access to captures for the URI-R.



(e) Private captures return to MMA, MA returns captures aggregated from three public Web archives.



(f) MMA aggregates results from a personal Web archive of public captures, a private Web archive, and an MA.

Fig. 77: An MMA may relay requests for captures from a set of Web archives instead of a single archive. This figure (Pattern 5) demonstrates a flow in aggregating captures from a private Web archive, personal Web archive with public captures, and three public Web archives via a conventional MA.

supports for aggregation $\{A_0\}$ and relays the `Prefer: privateOnly` supplied by Alice. The SG filters $\{A_0\}$ and provides the set of archives $\{A_f\}$ representative of only those that are private and thus meet the preference Alice specified. On the basis of this refined set of archives, the MMA can then perform the procedure described in Pattern 4 and Figure 75 where each respective private Web archive may require an authentication procedure and a respective PWAA.

For dimensions that require analysis of mementos' content to obtain resulting values, a StarGate may need to consult an additional service for this calculation. In Pattern 6b, Carol sends a request to an MMA with an HTTP request header of `Prefer: damage"<0.5"`. An MMA may recognize damage as a derived attribute and first send a request for a TimeMap for the URI-R Carol requested to each archive the MMA supports. Each archive is expected to return a TimeMap to the MMA, which the MMA aggregates and sends to a SG with the `Prefer: damage"<0.5"` header. The SG can then extract the URI-Ms from the aggregated TM. For each URI-M, the SG sends a request to a service to obtain a damage value that corresponds to this URI-M. With this set of corresponding values, the SG can then filter the URI-Ms based on the preference of `damage"<0.5"`. The SG generates a StarMap to associate mementos' respective URI-Ms with their damage values (and other available attributes like `datetime`) and prepends the StarMap with a metadata record like Figure 59. This StarMap is then returned to the MMA and relayed to Carol.

The third sub-pattern involves Alice again requesting captures from an MMA for a URI-R but specifying a content-based attribute using `Prefer: status="200"`. In scenarios like this compared to Pattern 6b, an SG does not consult an external service to analyze the memento by passing a URI-M as an argument. As with 6b, upon obtaining an aggregated TimeMap from an MMA, the MMA relays the TM to an SG. For each URI-M in the TimeMap, the SG sends a request to the URI-M (cf. a request to a service with the URI-M as an argument in 6b) and retains the status code of the memento. An SG may potentially cache this value, which we will explore further beyond this proposal (as described in Section 5.6.2). The SG can then filter the URI-Ms that meet the preference of the status code being `200` and generate a StarMap with this subset of URI-Ms, in a similar attribute association procedure and prepending as Pattern 6b. Likewise, the SM is returned to the MMA and from there returned to Alice.

5.4.7 PATTERN 7: AGGREGATION WITH FILTERING VIA SG INTERACTION

Each sub-pattern in Pattern 6 involved the client sending requests to an MMA. Pattern 7 details a client's interaction with a StarGate directly. Bob sends a request for a URI-R to a StarGate with `Prefer: damage"<0.5"` just as Carol did in Pattern 6b but to a different mementity. Bob also sends an additional `Prefer` request header specifying a custom set of archives he wants aggregated using the base64-encoding method described in Section 5.1.1. The SG that received Bob's request sends a request to an MMA with the set of archives Bob specified using the same `Prefer`-based mechanism and the URI-R. Using this set of archives, the MMA performs the aggregation procedure described in Pattern 2 and returns the StarMap to the SG. From this SM, the SG can then repeat the procedure similarly to how the mementity did when Carol requested captures with a preference in Pattern 6b. To accomplish this (as before) the SG sends a request to a damage calculation service with the respective URI-M as an argument. When the values are returned, the SG creates a SM only containing the mementos' identifiers and respective attributes that met the condition Bob specified, prepends the SM with the metadata information as in Pattern 6b, and returns the StarMap to Bob.

The set of access patterns can be summarized as follows. More patterns likely exist and will be explored during the course of the research.

Pattern 1: Single archive access

Pattern 2: Aggregation of multiple Web archives

Pattern 3: Aggregator chaining

Pattern 4: Aggregation with authentication

Pattern 5: Aggregation including a hybrid public-private archive

Pattern 6: Aggregation with filtering via MMA interaction

Pattern 7: Aggregation with filtering via SG interaction

5.5 FRAMEWORK EXTENSIBILITY

The mementities in Section 5.2 are designed to be applicable to a variety of existing user access patterns (some described in Section 5.4) with the intention of further applicability beyond the extent explored in this proposal. To ensure this, we have designed the mementities in this proposal to be functionally cohesive yet extensible. MMAs, for instance, contain the open-ended ability to interface with other mementities as they do with PWAAs (Section 5.4.4 and Figure 77), conventional Memento Aggregators (Section 5.4.3 and Figure 72), etc. Simultaneously, they offload the authentication and authorization process to an archive's respective PWAA, allowing each aggregated archive to retain their own authorization model so long as they return the result as modeled in this framework. By facilitating the applicability of the mementities to use cases beyond what we initially imagine, the potential reuse of the Mementity Framework both applied piecemeal (using a subset of mementities alone or in combination) and as a comprehensive hierarchy will also be facilitated.

Each mementity in Section 5.2 performed a single role in the hierarchical relation of each other respective mementity. In Section 5.4 we described seven access patterns, with the final five leveraging the new capability of the mementities for aggregating private and public Web archives. It is likely that other roles are necessary to account for the dynamics of some Web archives that are not yet covered in this preliminary work. In the event that this is needed, a mementity's role may be further refined or additional mementities introduced to allow for those that existed prior to and introduced in this proposal to remain functionally cohesive.

5.6 EVALUATION

Evaluation of this research will largely consist of determining the effectiveness of the hierarchy (graphically represented in Figure 69) in addressing the issues and research questions enumerated in Section 1.4. Further research is required in the sorts of access control needed in currently deployed private web archives that serve as barriers in protecting the content at the expense of integration with private web archives. The scalability of adding a layer of abstraction on top of currently deployed Memento aggregators will require concrete performance evaluation to determine how to effectively supplement the results aggregated from public web archives with those from private web archives. Quantitative success of the hierarchy can be tested when the scenarios described in Section 5.4 can be executed with the expected results returned. Correctness of the expected results will need to be determined to establish

a baseline to differentiate unexpected results and to account for variations in the fluctuating availability of various public and private web archives.

5.6.1 DESIGN DECISIONS

The role of each mementity in the Mementity Framework has been designed to initially cater to the user needs extrapolated from the Research Questions. It is likely that the design is not optimal, as real-world performance frequently informs subsequent optimizations of tools, frameworks, protocols, etc. While we have attempted to make the functionality of each mementity cohesive, it may be required that some additional functionality is subsumed or extracted to an additional mementity. For instance, the role of a PWAA of solely issuing and verifying tokens may also be needed to validate other forms of authentication and access based on privacy needs of an archive. We will analyze our original design decisions of the mementities in the framework, observe any inefficiencies in how the mementities interact, determine whether the roles are cohesive, and determine if any further optimizations are needed in the Mementity Framework. These optimizations will be performed to ensure applicability and usefulness for users who want to aggregate archives as well as allow the framework to be extended for currently unanticipated use cases.

5.6.2 COSTS OF GENERATING STARMAPS (AND LINK)

In Section 5.1.2 we discussed how adding additional attributes to mementos in TimeMaps (to produce StarMaps) and Link response headers makes them more expressive and useful. The procedure to obtain, process, and store these attributes will incur various costs to achieve. Spatial costs may be produced when storing StarMap variants (if permutations are stored), attributes (if in a database, implies temporal complexity to re-assemble), and calculated values (to prevent repeat incurrence). Temporal costs reside in the time required for requesting calculated attributes from external services and additional roundtrip time for the potentially necessary steps of a client requesting the supported attributes that can be used to enrich a TimeMap. We anticipate other computational costs to be incurred as well. Each of the costs will also need to be evaluated and abstracted to a computational complexity to account for the costs at-scale.

5.6.3 EVALUATION THROUGH IMPLEMENTATION

We anticipate both implementing the mementities in Section 5.2 either through extension of existing software (e.g., extending MemGator with the capabilities of an MMA) or new software packages. In addition to creating the mementities in the Mementity Framework, we plan to extend existing tools to utilize the mementities. One example that we will adapt as a reference implementation is to extend the Mink browser extension to serve as a user-accessible method of interacting with the mementities as well as take on some of the capabilities of an MMA (Section 5.3.3). This approach will provide a means for evaluating methods of expression for RQ4 and RQ5. Leveraging the browser, the tool that users use to access both the live and archived Webs, will provide a more realistic use case of the Mementity Framework. This will also facilitate further evaluation of the user-experience of the framework and allow us to more comprehensively answer RQ2.

To facilitate archive collaboration (Section 4.3) we will also extend either Mink and/or an additional browser-based tool to encourage sharing of personal and private captures and references to mementities (Section 5.3.4). We plan to use an approach similar to our work in creating InterPlanetary Wayback to integrate browser-based archival collaboration more seamlessly and distributed to account for the proliferation of personal Web archives. This approach at "Mink-to-Mink" communication may be able to utilize the work done with the JavaScript implementation of IPFS [139].

5.6.4 HOW WELL ARE CHAPTER 1 SCENARIOS AND ACCESS PATTERNS REALIZABLE

We plan to evaluate the level of resolution of the scenarios and issues described in Chapter 1. With the mementities of the Mementity Framework in-place, there will still be a question of ease-of-use with the reference implementations when they implement all features of the framework. This will likely influence the user interface decisions for integration to encourage the adoption of the framework through a user experience with minimal barriers. The mockup for archival selection for Mink in Figure 52 will be implemented and evaluated through user experience testing.

It is important to note that the Mementity Framework does not need to be comprehensively implemented to be of use. For example, if Carol wished to simply provide the ability to request a custom set of archives to be aggregated at the time of request by a client (Figure 53), only the MMA portion of the framework would be needed. In another fundamental use case, for Alice to not perform any aggregation or

negotiation to her private Web archive but still implement the authentication mechanism of the framework, she would only need to deploy a Private Web Archive Adapter (Figure 74). These two mementities and StarGates may be individually deployed for use but when implemented in combination, provide more of the capabilities of the framework to aggregate private and public Web archives.

One scenario described in Chapter 1 required the ability to distinguish captures with certain features. For example, Figure 10 showed two captures of the same URI, one of a personal and private representation and the other of a generic login page. Figure 7 shows multiple captures of `cnn.com` of a variety of qualities. Through surfacing these attributes (privacy and damage, respectively) and other dimensions through the Memento extension of StarMaps (Section 5.1) and being able to negotiate on these dimensions to obtain the aggregated result representative of URI-Ms that meet the conditions (as facilitated by the StarGate in Section 5.2.3), these scenarios in Chapter 1 may be considered resolved with the application of the Mementity Framework.

With a primary focus of the framework residing in the aggregation procedure, we plan to perform an extensive evaluation to determine the quantifiable effectiveness of the framework. We plan to reuse existing collections of publicly accessible Web archives to identify potentially sensitive or personal information. We plan to perform a collection procedure consisting of archives of URI-Rs of a variety of classes (example provided inline) as they correspond to the permutations of personal/institutional and public/private in Table I:

- public and well-archived (`cnn.com`)
- public but sparsely archived (`http://alicesembarassingphotos.net/vacation.html`)
- those that exhibit URI collision based on authentication status (`facebook.com`)
- those that are behind authentication from separate users (Alice's `facebook.com` vs. Carol's `facebook.com`)
- highly sensitive and behind authentication with an unlikelihood of URI collision (bank statements)

We plan to first separate out exemplars of each of these URI classes into buckets with the respective classification association retained. In a separate data set, we

plan to also co-mingle classes of URIs to use as a corpus for classification of potentially sensitive content that needs further consideration. From this we will produce a metric and method of classification that we can use on the aforementioned existing collections. Using the hierarchical aggregation approach introduced by adding an MMA, we will aggregate both our combined corpus, our class-based separated corpora, and the corpus as represented by existing archives. Through this aggregation we can evaluate the StarGate's functionality and effectiveness of negotiating on additional traits of the mementos contained within the data sets. For those data sets that we control (the combined and separated corpora), we can also evaluate the necessity and correctness of further authentication procedures, as may be facilitated by introducing a PWAA.

5.7 SUMMARY

In this chapter we defined the core mementities and fundamental dynamics of a Framework for aggregating private and public Web archives. In Section 5.1 we detailed different sorts of content negotiation that are needed, tailored, and represented for negotiation with Web archives in dimensions beyond time. Section 5.2 introduced three mementities (Memento Meta-Aggregator, a Private Web Archive Adapter, and a StarGate) and their roles and responsibilities as they pertain to aggregation, authentication, and negotiation to account for scenarios that arise when aggregating private and public Web archives. Section 5.3 described specific dynamics of the mementities and how they interact to form the hierarchical behavior of the framework. In Section 5.4 we built upon conventional access patterns to integrate usage of the mementities in Section 5.2, tying in the usage with real-world scenarios. In Section 5.5 we laid out the extensibility of the Mementity Framework to ensure that it is adaptable to unforeseen scenarios and dynamics in Web archiving.

CHAPTER 6

WORK SCHEDULE

In previous research, we have identified content that is problematic to preserve from the live Web (RQ1) [105], built tools to capture a portion of content that previously was not preserved (RQ2) [28, 107], and formulated a per-resource metric to evaluate the importance of content that is difficult to preserve (evaluation for RQ1) [46, 47]. Preserving this content is only a first step in replicating the archived Web experience in a manner that includes private and personal Web archives. The next steps we will perform will be to investigate modular and extensible approaches for access control when the nature of private Web archiving requires these considerations.

The entities built on top of the Memento framework (Section 5.2) will need to be scalable and account for unforeseen scenarios that will arise with the additional utilization of the infrastructure in-place to aggregate the archived public Web. Because our previous evaluations of resource importance [46, 47] only took into account public Web archives, the importance of resources captured from the private live Web will likely vary, so we will need to repeat experiments to evaluate the additional content that is much more difficult to capture.

Integrating public and private Web archives has an inherent problem of URI collision. A URI alone is an insufficient parameter (Section 1.2) for accessing content on parts of the private live Web. Replaying this content on the archived Web (containing both public and private live Web data) will require a deeper abstraction of access to reliably request content to replicate the experience from the live private Web (Section 5.3.1). Additional usage patterns (Section 5.4) not yet defined but to be explored in this research are very likely to exist when the additional entities for controlled access and aggregation are applied to real world scenarios. We will perform a user study for guidance in accounting for more of these situations.

The following sections in this chapter define concrete goals to be accomplished in our research. In Section 6.1 we explicitly define the research questions we will answer upon completion of the research outlined in this proposal. Section 6.2 highlights our current progress in the research including relevant peer-reviewed research we have published thus far.

6.1 RESEARCH QUESTIONS

Based on the issues previously enumerated, I wish to address the following research questions in the course of my thesis.

RQ1: What sort of content is difficult to capture and replay for preservation from the perspective of a web browser?

RQ2: How do Web browser APIs compare in potential functionality to the capabilities of archival crawlers?

RQ3: What issues exist for capturing and replaying content behind authentication?

RQ4: How can content that was captured behind authentication signal to Web archive replay systems that it requires special handling?

RQ5: How can Memento aggregators indicate that private Web archive content requires special handling to be replayed, despite being aggregated with publicly available web archive content?

RQ6: What kinds of access control do users who create private Web archives need to regulate access to their archives?

This research will be complete when the aforementioned goals toward preservation and access are attained. A timeline of prior, current, and upcoming progress will help steer the research to completion.

Prior publications in my doctoral research have dealt with the aspects of preservation [107, 102, 28, 48] (RQ1 and RQ2) and evaluation [46, 100, 105] (RQ1) aspects of the Framework. Future work will be focused on the access (RQ4 and RQ6) portion of the framework as well as expression of aggregation of public, private, personal, and institutional captures (RQ5).

6.2 TASKS AND PUBLICATIONS

We will consider the research described in this proposal complete when the tasks in Table VI have been completed. The relevancy of each task to our previous and anticipated peer-reviewed publications (described in Tables VII and VIII, respectively)

are also cross-referenced in Table VI. Figure 78 shows progress toward Table VI tasks along with other university requirements to consider the PhD complete.

ID	Task Name	RQ	Paper #
1	Study preservation of content behind authentication	1, 2	1 (JCDL 2012)
2	Study of Web archiving formats, software, practice, and personal and private Web Archiving	1, 2	1 (JCDL 2012), 3 (D-Lib 2013), 5 (JCDL 2014), 12 (JCDL 2017), 15 (JCDL 2018)
3	Archival Quality: How it has changed over time	1	2 (TPDL 2013)
4	Archival Quality: Current preservation tools' capabilities	1	5 (JCDL 2014), 15 (JCDL 2018)
5	Archival Quality: Importance of missing resources	1	6 (JCDL 2014), 7 (IJDL 2015), 8 (IJDL 2015)
6	Investigations in leveraging native browser APIs for personal Web archiving	2, 3	1 (JCDL 2012), 4 (JCDL 2014), 9 (JCDL 2015), 12 (JCDL 2017), 13 (JCDL 2017), 16 (JCDL 2018)
7	Proliferation and privacy of personal Web archives	4, 6	1 (JCDL 2012), 10 (JCDL 2016), 11 (TPDL 2016), 15 (JCDL 2018), *20*
8	Client-dictated Memento aggregation	5	17 (JCDL 2018), *18*
9	Content negotiation of Web archives on the dimension of personalization and privacy	4, 5, 6	17 (JCDL 2018), *19*

TABLE VI: Research progress relative to publications and research questions. Paper # corresponds to those in Table VII. Papers that are still to be published are listed in *italics*.

#	Title	Target/Venue	RQ	Status
1	WARCreate - Create Wayback-Consumable WARC Files from Any Webpage [107]	JCDL 2012	1, 2	Published
2	On the Change in Archivability of Websites Over Time [100]	TPDL 2013	1	Published
3	A Method for Identifying Personalized Representations in the Archives [99]	D-Lib Nov/Dec 2013	2	Published
4	Mink: Integrating the Live and Archived Web Viewing Experience Using Web Browsers and Memento [104]	JCDL 2014	2, 5	Published
5	The Archival Acid Test: Evaluating Archive Performance on Advanced HTML and JavaScript [105]	JCDL 2014	1	Published
6	Not All Mementos Are Created Equal: Measuring The Impact Of Missing Resources [46]	JCDL 2014	1	Published
7	The Impact of JavaScript on Archivability [48]	IJDL 2015	1	Published
8	Not All Mementos Are Created Equal: Measuring the Impact of Missing Resources [47]	IJDL 2015	1	Published
9	Mobile Mink: Merging Mobile and Desktop Archived Webs [89]	JCDL 2015	2	Published
10	InterPlanetary Wayback: The Permanent Web Archive [6]	JCDL 2016	4, 6	Published
11	InterPlanetary Wayback: Peer-To-Peer Permanence of Web Archives [95]	TPDL 2016	4, 6	Published
12	WAIL: Collection-Based Personal Web Archiving [28]	JCDL 2017	2	Published
13	Client-side Reconstruction of Composite Mementos Using ServiceWorker [7]	JCDL 2017	2	Published
14	Impact of URI Canonicalization on Memento Count [97]	JCDL 2017	3	Published
15	ArchiveNow: Simplified, Extensible, Multi-Archive Preservation [19]	JCDL 2018	1	Published
16	Unobtrusive and Extensible Archival Replay Banners Using Custom Elements [8]	JCDL 2018	1, 2	Published
17	A Framework for Aggregating Private and Public Web Archives [106]	JCDL 2018	3, 4, 5, 6	Published

TABLE VII: Research progress relative to peer-reviewed publications.

#	Title	Target/Venue	RQ	Status
18	Approaches Toward Client-Side Specification of Memento Aggregation		5	Planned
19	Negotiation of Web Archives Through Dimensions Beyond Time		6	Planned
20	A Survey of Private Web Archiving		1, 3, 6	Planned
21	A Framework for Aggregating Private and Public Web Archives	IJDL 2018/2019	1, 2, 3, 4, 5, 6	Planned

TABLE VIII: Anticipated peer-reviewed publications following my candidacy proposal. Indexes continued from Table VII.

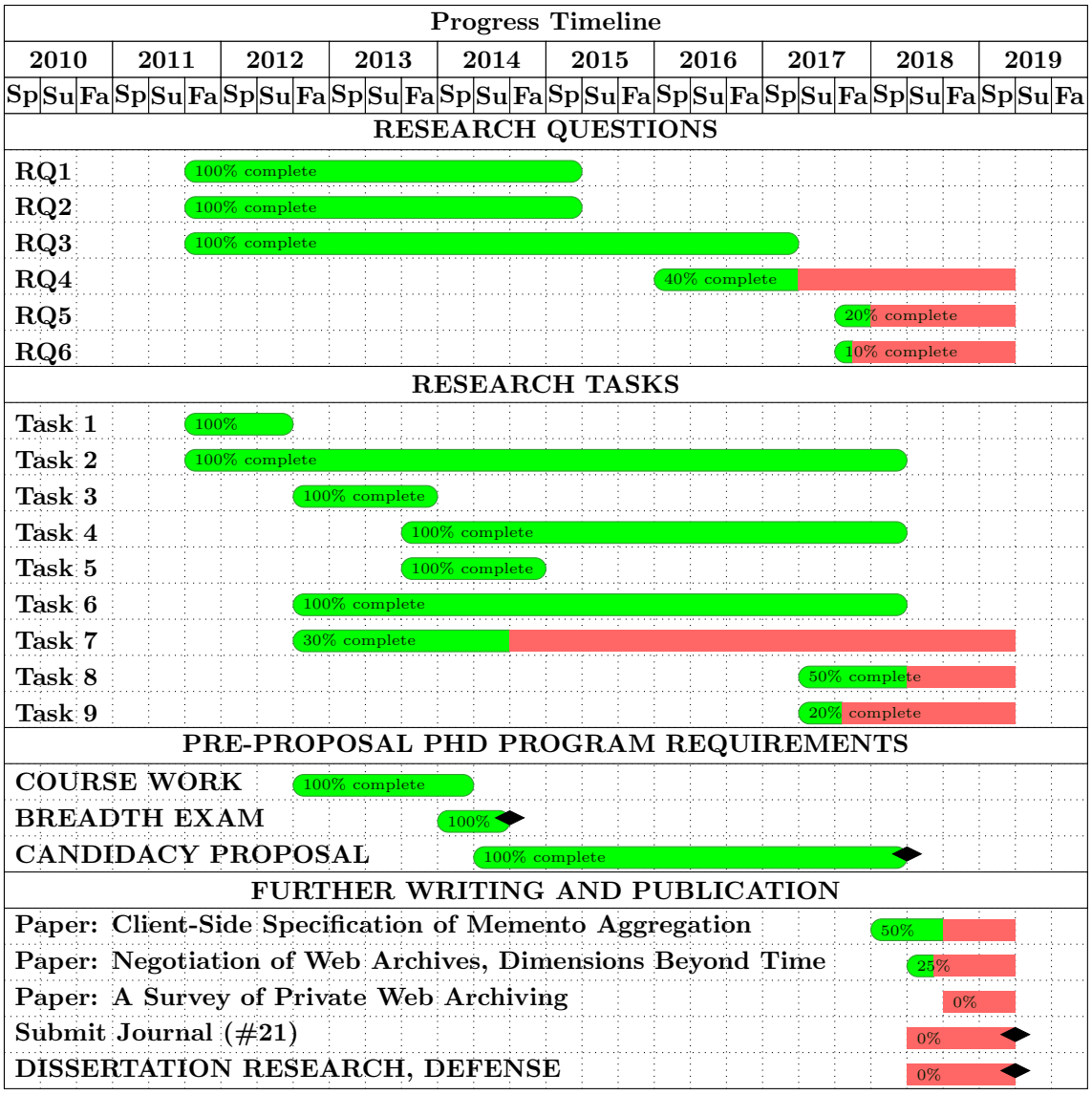


Fig. 78: Progress of answering Research Questions, completing Table VI tasks, as well as other requirements, complete and in-progress, with an anticipated timeline for completion.

REFERENCES

[1] Memento Tools: Proxy Scripts. http://www.mementoweb.org/tools/proxy/, May 2005.

[2] Memento Guide: Introduction. http://www.mementoweb.org/guide/quick-intro/, January 2015.

[3] David Abrams, Ron Baecker, and Mark Chignell. Information Archiving with Bookmarks: Personal Web Space Construction and Archiving. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 41–48, 1998.

[4] Scott G. Ainsworth, Ahmed Alsum, Hany SalahEldeen, Michele C. Weigle, and Michael L. Nelson. How Much of the Web is Archived? In *Proceeding of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 133–136, 2011.

[5] Sawood Alam. CDXJ: An Object Resource Stream Serialization Format. http://ws-dl.blogspot.com/2015/09/2015-09-10-cdxj-object-resource-stream.html, September 2015.

[6] Sawood Alam, Mat Kelly, and Michael L. Nelson. InterPlanetary Wayback: The Permanent Web Archive. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 273–274, 2016.

[7] Sawood Alam, Mat Kelly, Michele C. Weigle, and Michael L. Nelson. Client-side Reconstruction of Composite Mementos Using ServiceWorker. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 237–240, 2017.

[8] Sawood Alam, Mat Kelly, Michele C. Weigle, and Michael L. Nelson. Unobtrusive and Extensible Archival Replay Banners Using Custom Elements. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 319–320, June 2018.

[9] Sawood Alam and Michael L. Nelson. MemGator - A Portable Concurrent Memento Aggregator. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 243–244, 2016.

[10] Sawood Alam, Michael L. Nelson, Herbert Van de Sompel, Lyudmila L. Balakireva, Harihar Shankar, and David S. H Rosenthal. Web Archive Profiling Fulltext Search. In *Proceedings of the 20th International Conference on Theory and Practice of Digital Libraries (TPDL)*, pages 3–14, 2015.

[11] Sawood Alam, Michael L. Nelson, Herbert Van de Sompel, Lyudmila L. Balakireva, Harihar Shankar, and David S. H. Rosenthal. Web archive profiling through CDX summarization. *International Journal on Digital Libraries*, 17(3):223–238, 2016.

[12] Yasmin AlNoamany, Ahmed AlSum, Michele C. Weigle, and Michael L. Nelson. Who and What Links to the Internet Archive. In *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL)*, pages 346–357, 2013.

[13] Yasmin AlNoamany, Ahmed AlSum, Michele C. Weigle, and Michael L. Nelson. Who and What Links to the Internet Archive. *International Journal of Digital Libraries (IJDL)*, 14(3-4):101–115, 2014.

[14] Yasmin AlNoamany, Michele C. Weigle, and Michael L. Nelson. Access Patterns for Robots and Humans in Web Archives. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 339–348, 2013.

[15] Ahmed AlSum and Michael L. Nelson. Thumbnail Summarization Techniques for Web Archives. In *36th European Conference on IR Research (ECIR 2014)*, pages 299–310, 2014.

[16] Ahmed AlSum, Michele C. Weigle, Michael L. Nelson, and Herbert Van de Sompel. Profiling Web Archive Coverage for Top-Level Domain and Content Language. *International Journal on Digital Libraries*, 14(3-4):149–166, 2014.

[17] H. Alvestrand. Content Language Headers. IETF RFC 3282, May 2002.

[18] Grant Atkins. Paywalls in the Internet Archive. `http://ws-dl.blogspot.com/2018/03/2018-03-15-paywalls-in-internet-archive.html`, March 2018.

[19] Mohamed Aturban, Mat Kelly, Sawood Alam, John A. Berlin, Michael L. Nelson, and Michele C. Weigle. ArchiveNow: Simplified, Extensible, Multi-Archive

Preservation. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 321–322, 2018.

[20] Brenda Reyes Ayala, Mark E. Phillips, and Lauren Ko. Technical report, Current Quality Assurance Practices in Web Archiving, 2014. https://digital.library.unt.edu/ark:/67531/metadc333026/.

[21] Jefferson Bailey, Abigail Grotke, Kristine Hanna, Cathy Hartman, Edward McCain, Christie Moffatt, and Nicholas Taylor. Web Archiving in the United States: A 2013 Survey. Technical report, The National Digital Stewardship Alliance. http://ndsa.org/documents/NDSA_USWebArchivingSurvey_2013.pdf.

[22] Jefferson Bailey, Abigail Grotke, Edward McCain, Christie Moffatt, and Nicholas Taylor. Web Archiving in the United States: A 2016 Survey. Technical report, The National Digital Stewardship Alliance. http://ndsa.org/documents/WebArchivingintheUnitedStates_A2016Survey.pdf.

[23] A. Barth. HTTP State Management Mechanism. IETF RFC 6265, April 2011.

[24] M. Belshe, R. Peon, and M. Thomson. Hypertext Transfer Protocol Version 2 (HTTP/2). IETF RFC 7540, May 2015.

[25] Anat Ben-David and Hugo Huurdeman. Web Archive Search as Research: Methodological and Theoretical Implications. *Alexandria*, 25(1-2):93–111, 2014.

[26] John Berlin. CNN.com has been unarchivable since November 1st, 2016. http://ws-dl.blogspot.com/2017/01/2017-01-20-cnncom-has-been-unarchivable.html, January 2017.

[27] John Berlin. To Relive The Web: A Framework for the Transformation and Archival Replay of Web Pages. Master's thesis, Old Dominion University, Norfolk, Virginia, USA, 2018.

[28] John A. Berlin, Mat Kelly, Michael L. Nelson, and Michele C. Weigle. WAIL: Collection-Based Personal Web Archiving. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 340–341, 2017.

[29] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext Transfer Protocol – HTTP/1.0. IETF RFC 1945, May 1999.

[30] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifiers (URI): Generic Syntax. IETF RFC 2396, August 1998.

[31] T. Berners-Lee, R. Fielding, and L. Massinter. Uniform Resource Identifier (URI): Generic Syntax, Internet RFC-3986. IETF RFC 3986, January 2005.

[32] Tim Berners-Lee. Information Management: A Proposal. 1989. `https://www.w3.org/History/1989/proposal.html`.

[33] Tim Berners-Lee. Web Architecture: Generic Resources. 1996. `http://www.w3.org/DesignIssues/Generic.html`.

[34] Tim Berners-Lee. Cool URIs don't change. 1998. `http://www.w3.org/Provider/Style/URI.html`.

[35] Tim Berners-Lee. Universal Resource Identifiers – Axioms of Web Architecture. 1996 December. `https://www.w3.org/DesignIssues/Axioms.html`.

[36] Tim Berners-Lee. What do HTTP URIs Identify? 2002 July. `https://www.w3.org/DesignIssues/HTTP-URI.html`.

[37] N. Borenstein and N. Freed. MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies. IETF RFC 1521, September 1993.

[38] Nicolas J. Bornand, Lyudmila Balakireva, and Herbert Van de Sompel. Routing Memento Requests Using Binary Classifiers. In *Proceedings of the ACM/IEEE-CS on Joint Conference on Digital Libraries (JCDL)*, pages 63–72, 2016.

[39] R. Braden. Requirements for Internet Hosts – Application and Support. IETF RFC 1123, October 1989.

[40] Brave Software Inc. Brave Software — Building a Better Web. `https://brave.com/`.

[41] T. Bray. The JavaScript Object Notation (JSON) Data Interchange Format. IETF RFC 7159, March 2014.

[42] T. Bray. An HTTP Status Code to Report Legal Obstacles. IETF RFC 7725, February 2016.

[43] Brewster Kahle. Wayback Machine update +4Billion to 658,661,007,000 web objects archived and served. (small indexes update with recent captures, then a big sweep into a big update, like this one). 658Billion! go @internetarchive go @waybackmachine ! `https://twitter.com/brewster_kahle/status/1016003169589981184`, July 2018.

[44] Justin F. Brunelle. *Scripts in a Frame: A Framework for Archiving Deferred Representations*. PhD thesis, Old Dominion University Department of Computer Science, 2016.

[45] Justin F. Brunelle, Krista Ferrante, Eliot Wilczek, Michele C. Weigle, and Michael L. Nelson. Leveraging Heritrix and the Wayback Machine on a Corporate Intranet: A Case Study on Improving Corporate Archives. *D-Lib Magazine*, 22(1/2), 2016.

[46] Justin F. Brunelle, Mat Kelly, Hany SalahEldeen, Michele C. Weigle, and Michael L. Nelson. Not All Mementos Are Created Equal: Measuring The Impact Of Missing Resources. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 321–330, 2014.

[47] Justin F. Brunelle, Mat Kelly, Michele C. Weigle, and Michael L. Nelson. Not All Mementos Are Created Equal: Measuring the Impact of Missing Resources. *International Journal on Digital Libraries*, 16(3):283–301, 2015.

[48] Justin F. Brunelle, Mat Kelly, Michele C. Weigle, and Michael L. Nelson. The Impact of JavaScript on Archivability. *International Journal on Digital Libraries*, 17(2):95–117, 2016.

[49] Justin F. Brunelle and Michael L. Nelson. An Evaluation of Caching Policies for Memento TimeMaps. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 267–276, 2013.

[50] Justin F. Brunelle, Michele C. Weigle, and Michael L. Nelson. Archival Crawlers and JavaScript: Discover More Stuff but Crawl More Slowly. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 1–10, 2017.

[51] Mike Burner and Brewster Kahle. Arc File Format. http://archive.org/web/researcher/ArcFileFormat.php, September 1996.

[52] Moses S. Charikar. Similarity Estimation Techniques from Rounding Algorithms. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing (STOC'02)*, pages 380–388, 2002.

[53] Wendy Chisholm, Gregg Vanderheiden, and Ian Jacobs. Web Content Accessibility Guidelines 1.0. *Interactions*, 8(4):35–54, July 2001.

[54] Miguel Costa and Mário J Silva. Understanding the Information Needs of Web Archive Users. In *Proceedings of the 10th International Web Archiving Workshop (IWAW'10)*, pages 9–16, 2010.

[55] Michael Day. Collecting and preserving the World Wide Web. Technical report, Joint Information Systems Committee (JISC), 2003. http://library.wellcome.ac.uk/assets/WTL039229.pdf.

[56] Alexis Deveria. Can I use... Support tables for HTML5, CSS3, etc - File API, 2017. https://caniuse.com/#feat=fileapi.

[57] Thomas E. Dickey. Lynx – The Text Web-Browser. http://lynx.invisible-island.net/.

[58] Dalibor D. Dvorski. Installing, Configuring, and Developing with XAMPP. Skills Canada, 2007.

[59] D. Eastlake and A. Panitz. Reserved Top Level DNS Names. IETF RFC 2606, June 1999.

[60] Gunther Eysenbach and Mathieu Trudel. Going, Going, Still There: Using the WebCite Service to Permanently Archive Cited Web Pages. *Journal of Medical Internet Research*, 7(5), 2005.

[61] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. IETF RFC 2616, May 1996.

[62] R. Fielding, Y. Lafon, and J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Range Requests. IETF RFC 7233, June 2014.

[63] R. Fielding, M. Nonntingham, and J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Caching. IETF RFC 7234, June 2014.

[64] R. Fielding and J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Authentication. IETF RFC 7235, June 2014.

[65] R. Fielding and J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests. IETF RFC 7232, June 2014.

[66] R. Fielding and J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing. IETF RFC 7230, June 2014.

[67] R. Fielding and J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. IETF RFC 7231, June 2014.

[68] J.J. Garrett. Ajax: A New Approach to Web Applications. `http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications`, 2005.

[69] Vinay Goel. Web Archive Analysis. Presented at Archive-It Partner Meeting, 2013.

[70] Y. Goland, E. Whitehead, A. Faizi, S. Carter, and D. Jensen. HTTP Extensions for Distributed Authoring – WEBDAV. IETF RFC 2518, February 1999.

[71] Daniel Gomes, Sérgio Freitas, and Mário J Silva. Design and Selection Criteria for a National Web Archive. In *Research and Advanced Technology for Digital Libraries*, pages 196–207. Springer, 2006.

[72] Daniel Gomes, João Miranda, and Miguel Costa. A Survey on Web Archiving Initiatives. In *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL)*, pages 408–420, 2011.

[73] J. Gregorio and B. de hOra. The Atom Publishing Protocol. IETF RFC 5023, December 2007.

[74] Dick Hardt. The OAuth 2.0 Authorization Framework. IETF RFC 6749, October 2012.

[75] Philippe Le Hégaret, Ray Whitmer, and Lauren Wood. Document Object Model, January 2005. `https://www.w3.org/DOM/`.

[76] Ian Hickson, Robin Berjon, Steve Faulkner, Travis Leithead, Erika Doyle Navara, Edward O'Connor, and Silvia Pfeiffer. Document Metadata — HTML5, October 2014.

[77] K. Holtman and A. Mutz. Transparent Content Negotiation in HTTP. IETF RFC 2295, March 1998.

[78] Hugo C. Huurdeman, Anat Ben-David, and Thaer Sammar. Sprint Methods for Web Archive Research. In *Proceedings of the 5th Annual ACM Web Science Conference*, pages 182–190, 2013.

[79] International Internet Preservation Consortium. OpenWayback. `https://github.com/iipc/openwayback`, 2014.

[80] Internet Archive. Removing Documents From the Wayback Machine. `http://web.archive.org/web/20151031123632/https://archive.org/about/exclude.php`, March 2015.

[81] Internet Assigned Numbers Authority (IANA). Link Relation. `https://www.iana.org/assignments/link-relations/`, August 2005.

[82] ISO 28500. WARC (Web ARChive) file format. `http://www.digitalpreservation.gov/formats/fdd/fdd000236.shtml`, August 2009.

[83] Andrew N. Jackson. The CDX file format (2015). `https://iipc.github.io/warc-specifications/specifications/cdx-format/cdx-2015/`.

[84] Andrew N. Jackson. Can a Web Archive Lie? `http://anjackson.net/2017/06/29/waw-your-lying-archives/`, June 2017.

[85] Ian Jacobs and Norman Walsh. Architecture of the world wide web, volume one. Technical Report W3C Recommendation 15 December 2004, W3C, 2004.

[86] Mike Jones and Dick Hardt. The OAuth 2.0 Authorization Framework: Bearer Token Usage. IETF RFC 6750, October 2012.

[87] Shawn M. Jones, Herbert Van de Sompel, and Michael L. Nelson. Mementos in the Raw. http://ws-dl.blogspot.com/2016/04/2016-04-27-mementos-in-raw.html, April 2016.

[88] Shawn M. Jones, Herbert Van de Sompel, Harihar Shankar, Martin Klein, Richard Tobin, and Claire Grover. Scholarly Context Adrift: Three out of Four URI References Lead to Changed Content. *PLOS ONE*, 11(12):1–32, 12 2016.

[89] Wesley Jordan, Mat Kelly, Justin F. Brunelle, Laura Vobrak, Michele C. Weigle, and Michael L. Nelson. Mobile Mink: Merging Mobile and Desktop Archived Webs. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 243–244, 2015.

[90] Dhaval Kapil. Attacking the OAuth Protocol. https://dhavalkapil.com/blogs/Attacking-the-OAuth-Protocol/, February 2017.

[91] Mat Kelly. An Extensible Framework for Creating Personal Archives of Web Resources Requiring Authentication. Master's thesis, Old Dominion University, Norfolk, Virginia, USA, 2012.

[92] Mat Kelly. A Framework for Aggregating Private and Public Web Archives. *Bulletin of the IEEE Technical Committee on Digital Libraries (TCDL)*, 11(3), 2015.

[93] Mat Kelly. Facilitation of the A Posteriori Replication of Web Published Satellite Imagery. Virginia Space Grant Consortium 2015 Student Research Conference, April 2015. http://www.cs.odu.edu/~mkelly/papers/2015_vsgc_imagery.pdf.

[94] Mat Kelly. 2016-06-03: Lipstick or Ham: Next Steps for WAIL. http://ws-dl.blogspot.com/2016/06/2016-06-03-lipstick-or-ham-next-steps.html, June 2016.

[95] Mat Kelly, Sawood Alam, Michael L. Nelson, and Michele C. Weigle. InterPlanetary Wayback: Peer-To-Peer Permanence of Web Archives. In *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL)*, pages 411–416, 2016.

[96] Mat Kelly, Sawood Alam, Michael L. Nelson, and Michele C. Weigle. Client-Assisted Memento Aggregation Using the Prefer Header. In *Web Archiving and Digital Libraries (WADL) Workshop*, 2018. https://fox.cs.vt.edu/wadl2018.html.

[97] Mat Kelly, Lulwah M. Alkwai, Sawood Alam, Michael L. Nelson, Michele C. Weigle, and Herbert Van de Sompel. Impact of URI Canonicalization on Memento Count. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 303–304, 2017.

[98] Mat Kelly, Lulwah M Alkwai, Michael L. Nelson, Michele C. Weigle, and Herbert Van de Sompel. Impact of URI Canonicalization on Memento Count. Technical Report arXiv:1703.03302, Old Dominion University, March 2017.

[99] Mat Kelly, Justin F. Brunelle, Michele C. Weigle, and Michael L. Nelson. A Method for Identifying Personalized Representations in the Archives. *D-Lib Magazine*, 19(11/12), Nov/Dec 2013.

[100] Mat Kelly, Justin F. Brunelle, Michele C. Weigle, and Michael L. Nelson. On the Change in Archivability of Websites Over Time. In *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL)*, pages 35–47, 2013.

[101] Mat Kelly and David Dias. A Collaborative, Secure, and Private InterPlanetary Wayback Archiving System using IPFS. Presented at International Internet Preservation Consortium (IIPC) Web Archiving Conference (WAC) 2017, June 2017. https://www.slideshare.net/machawk1/a-collaborative-secure-and-private-interplanetary-wayback-web-archiving-system-using-ipfs.

[102] Mat Kelly, Michael L. Nelson, and Michele C. Weigle. Making Enterprise-Level Archive Tools Accessible for Personal Web Archiving Using XAMPP. Presented at Personal Digital Archiving, February 2013. http://www.cs.odu.edu/~mkelly/posters/2013_pda_wail.pdf.

[103] Mat Kelly, Michael L. Nelson, and Michele C. Weigle. WARCreate and WAIL: WARC, Wayback, and Heritrix Made Easy. Presented at Web Archiving Workshop at Digital Preservation 2013,

July 2013. http://www.cs.odu.edu/~mkelly/presentations/2013_digitalPreservation_heritrixMadeEasy.pptx.

[104] Mat Kelly, Michael L. Nelson, and Michele C. Weigle. Mink: Integrating the Live and Archived Web Viewing Experience Using Web Browsers and Memento. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 469–470, 2014.

[105] Mat Kelly, Michael L. Nelson, and Michele C. Weigle. The Archival Acid Test: Evaluating Archive Performance on Advanced HTML and JavaScript. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 25–28, 2014.

[106] Mat Kelly, Michael L. Nelson, and Michele C. Weigle. A Framework for Aggregating Private and Public Web Archives. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 273–282, June 2018.

[107] Mat Kelly and Michele C. Weigle. WARCreate - Create Wayback-Consumable WARC Files from Any Webpage. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 437–438, 2012.

[108] Mat Kelly, Michele C. Weigle, and Michael L. Nelson. Archiving Your Facebook Pages Using Archive Facebook. NDIIPP/NDSA Partners Meeting Special Interest Session, July 2011. http://www.cs.odu.edu/~mkelly/presentations/2011_ndiipp_archivefacebook.pptx.

[109] Mat Kelly, Michele C. Weigle, and Michael L. Nelson. WARCreate - Create Wayback-Consumable WARC Files from Any Webpage. Presented at Web Archiving Workshop at Digital Preservation 2012, July 2012. http://www.cs.odu.edu/~mkelly/presentations/2012_digitalPreservation_warcreate.pptx.

[110] David S. Kirk and Abigail Sellen. On Human Remains: Values and Practice in the Home Archiving of Cherished Objects. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 17(3):10, 2010.

[111] Ilya Kreymer. webrecorder/pywb: Core Python Web Archiving Toolkit for replay and recording of web archives. https://github.com/webrecorder/pywb, 2016.

[112] D. Kristol and L. Montulli. HTTP State Management Mechanism. IETF RFC 2109, February 1997.

[113] D. Kristol and L. Montulli. HTTP State Management Mechanism. IETF RFC 2965, October 2000.

[114] Rhys Lewis. Dereferencing HTTP URIs. May 2007. `https://www.w3.org/2001/tag/doc/httpRange-14/2007-05-31/HttpRange-14`.

[115] Evan Light. The Snowden Archive-in-a-Box: A year of travelling experiments in outreach and education. *Big Data & Society*, 3(2):1–7, 2016.

[116] Siân E. Lindley, Catherine C. Marshall, Richard Banks, Abigail Sellen, and Tim Regan. Rethinking the Web as a Personal Archive. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 749–760, 2013.

[117] Catherine C. Marshall. Rethinking Personal Digital Archiving, Part 1. *D-Lib Magazine*, 14(3/4), Mar/Apr 2008.

[118] Catherine C. Marshall. Rethinking Personal Digital Archiving, Part 2. *D-Lib Magazine*, 14(3/4), Mar/Apr 2008.

[119] Catherine C Marshall and Frank M Shipman. The Ownership and Reuse of Visual Media. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 157–166, 2011.

[120] Catherine C. Marshall and Frank M. Shipman. On the Institutional Archiving of Social Media. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 1–10, 2012.

[121] Catherine C. Marshall and Frank M. Shipman. An Argument for Archiving Facebook as a Heterogeneous Personal Store. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 11–20, 2014.

[122] Julien Masanès. Web Archiving: Issues and Methods. In *Web Archiving*, pages 1–53. Springer, 2006.

[123] Frank McCown, Catherine C. Marshall, and Michael L. Nelson. Why Websites Are Lost (and How They're Sometimes Found). *Communications of the ACM*, 52(11):141–145, 2009.

[124] Frank McCown and Michael L. Nelson. What Happens When Facebook is Gone? In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 251–254, 2009.

[125] Jim McGrath, Alicia Peaker, Ryan Cordell, and Elizabeth Maddock Dillon. Our Marathon: The Boston Bombing Digital Archive. `http://marathon.neu.edu/`, 2013-2015.

[126] MITRE Corporation. FFRDCs - A Primer. `https://www.mitre.org/sites/default/files/publications/ffrdc-primer-april-2015.pdf`, April 2015.

[127] Gordon Mohr, Michael Stack, Igor Ranitovic, Dan Avery, and Michele Kimpton. An Introduction to Heritrix, An open source archival quality web crawler. In *4th International Web Archiving Workshop (IWAW'04)*, September 2004.

[128] NDSA Content Working Group. Web Archiving Survey Report. Technical report, The National Digital Stewardship Alliance. `http://www.digitalpreservation.gov/documents/ndsa_web_archiving_survey_report_2012.pdf`.

[129] Jinfang Niu. Functionalities of Web Archives. *D-Lib Magazine*, 18(3/4), Mar/Apr 2012.

[130] M. Nottingham. Web Linking. IETF RFC 5988, October 2010.

[131] M. Nottingham. URI Design and Ownership. IETF RFC 7320, July 2014.

[132] M. Nottingham. Web Linking. IETF RFC 8288, October 2017.

[133] M. Nottingham. HTTP Representation Variants. IETF RFC draft, June 2018.

[134] M. Nottingham and R. Sayre. The Atom Syndication Format. IETF RFC 4287, December 2005.

[135] Pandora Archive. PADI : preserving access to digital information (including ICADS). `http://pandora.nla.gov.au/tep/10691`, August 2011.

[136] Margaret Phillips. PANDORA, Australia's Web Archive, and the Digital Archiving System that Supports It. *National Library of Australia Staff Papers*, 2009.

[137] Liza Potts. Social Media Systems in Times of Disaster: Users Becoming Participants. *UXPA Magazine*, 15(1), 2015.

[138] Liza Potts, Joyce Seitzinger, Dave Jones, and Angela Harrison. Tweeting Disaster: Hashtag Constructions and Collisions. In *Proceedings of the 29th ACM International Conference on Design of Communication*, pages 235–240, 2011.

[139] Protocol Labs. ipfs/js-ipfs: IPFS implementation in JavaScript. `https://github.com/ipfs/js-ipfs`, 2017.

[140] Arun Ranganathan and Jonas Sicking. File API. *W3C*, 2015. `https://www.w3.org/TR/FileAPI/`.

[141] Herman Chung-Hwa Rao, Yih-Farn Chen, and Ming-Feng Chen. A Proxy-based Personal Web Archiving Service. *SIGOPS Operating System Review*, 35(1):61–72, January 2001.

[142] Andreas Rauber, Max Kaiser, and Bernhard Wachter. Ethical Issues in Web Archive Creation and Usage-Towards a Research Agenda. In *8th International Web Archiving Workshop (IWAW'08)*, 2008.

[143] Rhizome. Webrecorder. `https://webrecorder.io`, 2017.

[144] J. Rosenberg. What's in a Name: False Assumptions about DNS Names. IETF RFC 4367, February 2006.

[145] David S. H. Rosenthal. The Importance of Discovery in Memento. `http://blog.dshr.org/2010/12/importance-of-discovery-in-memento.html`, December 2010.

[146] David S. H. Rosenthal. Memento & the Marketplace for Archiving. `http://blog.dshr.org/2011/01/memento-marketplace-for-archiving.html`, January 2011.

[147] David S. H. Rosenthal. Re-thinking Memento Aggregation. `http://blog.dshr.org/2013/03/re-thinking-memento-aggregation.html`, March 2013.

[148] David S. H. Rosenthal. Content negotiation and Memento. http://blog.dshr.org/2016/08/content-negotiation-and-memento.html, August 2016.

[149] David S. H. Rosenthal, Thomas Robertson, Tom Lipkis, Vicky Reich, and Seth Morabito. Requirements for Digital Preservation Systems. *D-Lib Magazine*, 11(11), November 2005.

[150] Alexis Rossi. Fixing Broken Links on the Internet — Internet Archive Blogs. https://blog.archive.org/2013/10/25/fixing-broken-links/, October 2013.

[151] Jeff Rothenberg. Ensuring the Longevity of Digital Information. *International Journal ofLegal Information*, 26(1):1–18, 1998.

[152] M-J. Saarinen and J-P. Aumasson. The BLAKE2 Cryptographic Hash and Message Authentication Code (MAC). IETF RFC 7693, November 2015.

[153] P. Saint-Andre, D. Crocker, and M. Nottingham. Deprecating the "X-" Prefix and Similar Constructs in Application Protocols. IETF RFC 6648, June 2012.

[154] Leo Sauermann and Richard Cyganiak. Cool URIs for the semantic web. Technical Report W3C Interest Group Note 31 March 2008, W3C, 2008.

[155] Thomas Schwarz, Mary Baker, Steven Bassi, Bruce Baumgart, Wayne Flagg, Catherine van Ingen, Kobus Joste, Mark Manasse, and Mehul Shah. Disk Failure Investigations at the Internet Archive. In *Work-in-Progess session, NASA/IEEE Conference on Mass Storage Systems and Technologies (MSST2006)*, 2006.

[156] Z. Shelby. Constrained RESTful Environments (CoRE) Link Format. IETF RFC 6690, August 2012.

[157] K. Sigurðsson. Incremental Crawling with Heritrix. In *Proceedings of the 5th International Web Archiving Workshop (IWAW'05)*, 2005.

[158] K. Sigurðsson, M. Stack, and I. Ranitovic. Heritrix User Manual: Sort-friendly URI Reordering Transformation, 2006. http://crawler.archive.org/articles/user_manual/glossary.html#surt.

[159] D. Singer, R. Clark, and D. Lee. MIME Type Registrations for JPEG 2000 (ISO/IEC 15444). IETF RFC 3745, April 2004.

[160] J. Snell. Prefer Header for HTTP. IETF RFC 7240, June 2014.

[161] Kathryn Stine. Cobweb: Collaborative Collection Development for Web Archives. `https://www.cdlib.org/services/cobweb/`, June 2018.

[162] Stephan Strodl, Florian Motlik, Kevin Stadler, and Andreas Rauber. Personal & Soho Archiving. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 115–123, 2008.

[163] Aaron Swartz. Aaron Swartz's A Programmable Web: An Unfinished Work. *Synthesis Lectures on The Semantic Web: Theory and Technology*, 3(2):1–64, 2013.

[164] Mike Thelwall and Liwen Vaughan. A fair history of the Web? Examining country balance in the Internet Archive. *Library & Information Science Research*, 26(2):162–176, 2004.

[165] Brad Tofel. 'Wayback' for Accessing Web Archives. In *7th International Web Archiving Workshop (IWAW'07)*, 2007.

[166] Heather Tweedy, Frank McCown, and Michael L. Nelson. A Memento Web Browser for iOS. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 371–372, 2013.

[167] UK Web Archive. ukwa/ukwa-pywb. `https://github.com/ukwa/ukwa-pywb`, 2018.

[168] United States Access Board. The Rehabilitation Act Amendments (Section 508). `https://www.access-board.gov/sec508/`, 1998.

[169] Herbert Van de Sompel, Michael Nelson, and Robert Sanderson. HTTP Framework for Time-Based Access to Resource States – Memento. IETF RFC 7089, December 2013.

[170] Herbert Van de Sompel, Michael L. Nelson, Lyudmila Balakireva, Martin Klein, Shawn M. Jones, and Harihar Shankar. Mementos In the

Raw, Take Two. http://ws-dl.blogspot.com/2016/08/2016-08-15-mementos-in-raw-take-two.html, August 2016.

[171] Ting Wang, Mudhakar Srivatsa, and Ling Liu. Fine-Grained Access Control of Personal Data. In *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*, pages 145–156, 2012.

[172] Jason Webber. A New Playback Tool for the UK Web Archive - UK Web Archive blog. http://blogs.bl.uk/webarchive/2018/02/a-new-playback-tool-for-the-uk-web-archive.html, February 2018.

[173] Michele C. Weigle, Michael L. Nelson, and Liza Potts. "Archive What I See Now": Bringing Institutional Web Archiving Tools to the Individual Researcher. http://ws-dl.blogspot.com/2014/07/2014-07-22-archive-what-i-see-now.html, July 2014.

[174] E. Wilde. The 'profile' Link Relation Type. IETF RFC 6906, March 2013.

[175] World Wide Web Consortium (W3C). Web Accessibility Initiative (WAI). Technical report. https://www.w3.org/WAI/.