AGGREGATING PRIVATE AND PUBLIC WEB ARCHIVES USING THE MEMENTITY FRAMEWORK

by

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

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
August 2019

ABSTRACT

AGGREGATING PRIVATE AND PUBLIC WEB ARCHIVES USING THE MEMENTITY FRAMEWORK

Matthew R. Kelly
Old Dominion University, 2019
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 archived pages by individuals and organizations of personal, private, and public Web content. This is misrepresentative of the Web as it was. While the Memento Framework 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 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 evaluate the framework by justifying the mementity design decisions, formulaically abstracting the anticipated temporal and spatial costs, and providing reference implementations, usage, and examples for the framework.

To Scarlett and Melissa, my Figure 1s.

ACKNOWLEDGEMENTS

I would like to thank my committee members for their willingness to review my dissertation and for their valuable feedback during The Process. I would like to particularly thank Dr. Michele C. Weigle and Dr. Michael L. Nelson for their iterative, weekly, meticulous feedback on both the written content as well as the concepts described therein. A third thanks goes to Dr. Weigle for her willingness to take a chance on me when first joining ODU.

I would also like to thank all of my lab mates at the Web Science and Digital Libraries Research Group, who each contributed to the completion in their own way. For early reviews I would like to thank Scott G. Ainsworth and Shawn M. Jones. For seeding the ideas and allowing for the initial exploration of the framework concepts, I would like to thank Justin F. Brunelle and Hany SalahEldeen. For advanced insights and exploration of some novel dynamics in my research, I would like to thank John A. Berlin. Sawood Alam also provided a great deal of open-ended feedback and advice on how to go forward in the exploration and for that I am extremely appreciative.

A few individuals provided very early facilitation of completion or supported me in ways that would surprise them to know contributed to a PhD. For these contributions I would like to thank Ryan Glaser, Sue Schaeffer, Barbara Mehaffey, Charles Steadham, Brian Conner, and Andrew Bobrowski.

I would like to thank my family for support. I would like to particularly thank my mother, Michele Kelly, for providing motivation and accountability along the way; my father, Kevin Kelly, for providing the rationale and sound advice I needed through the slog; and my younger brother, Dr. Sean B. Kelly, for poking and prodding at the dissertation concepts to provide sanity checks and ensure its merit.

Most importantly, I would like to thank my wife, Melissa, for enduring our trek and my frustrations in taking a long path in my research. Her tolerance, support, and reassurance were critical to completion.

Finally, I would like the thank the funding agencies that supported my PhD research. For indirect support I would like to thank the National Science Foundation Division Of Computer and Network Systems 1116238 and Institute of Museum and Library Services RE-33-16-0107-16, which both allowed for open-ended exploration of my early dissertation work. I would also like to thank the National Endowment for the Humanities HK-50181-14 for funding the advancement of very specific parts of the framework and seeing potential in my programmatic products. Some figures and use cases in this dissertation are expressed using work derived from Agata Krych, licensed as CC BY-SA, and available at `https://matkelly.com/dissertation`.

This list is not comprehensive, though, I want to again thank those that are listed and others that are not. I am very grateful for your support in the completion of my PhD dissertation.

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.

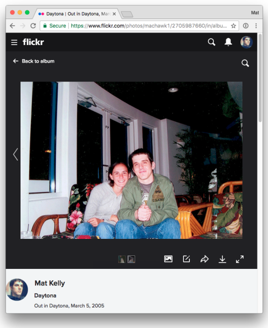
- Jeff Rothenberg, *Ensuring the Longevity of Digital Information [177]*

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 [134].

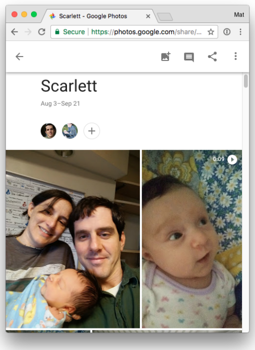
[1]`https://facebook.com`
[2]`https://photos.google.com`

(a)

(b)



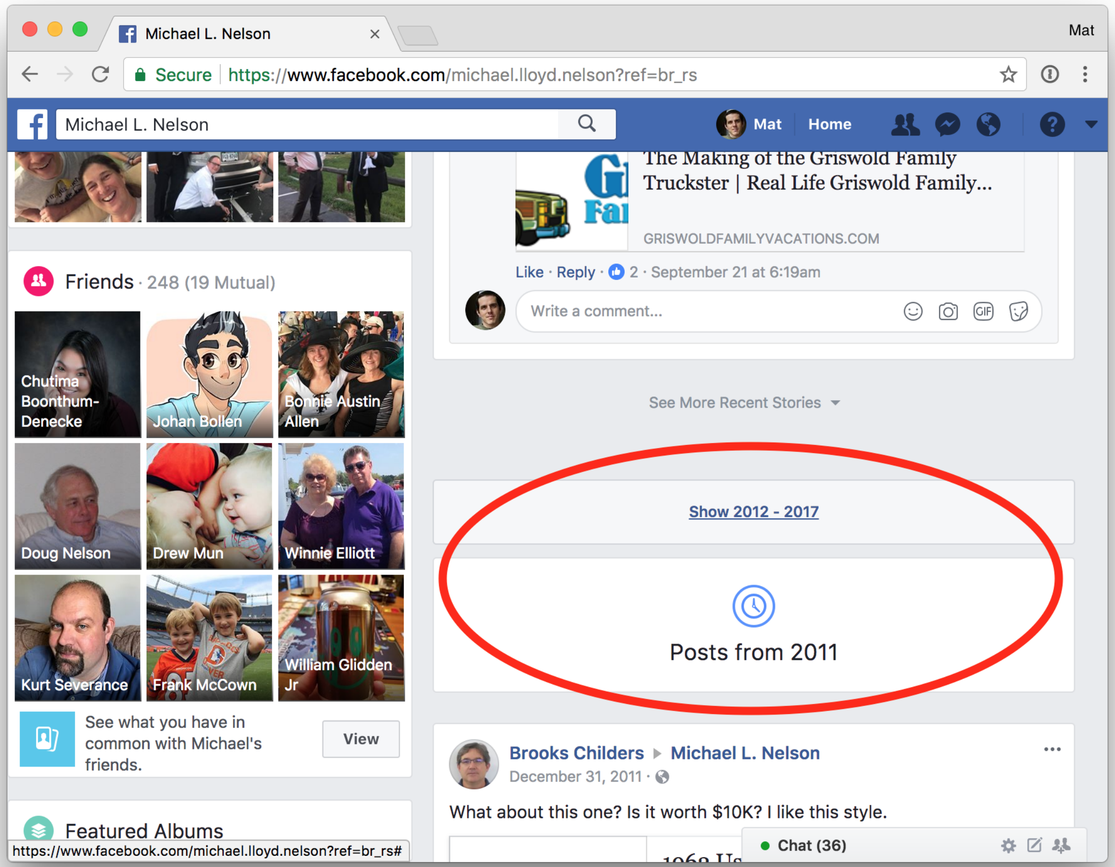
(c)



(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 [134, 128], 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).



https://www.facebook.com/michael.lloyd.nelson?ref=br_rs

The Making of the Griswold Family Truckster | Real Life Griswold Family...
GRISWOLDFAMILYVACATIONS.COM

See what you have in common with Michael's friends.

Brooks Childers ▸ Michael L. Nelson
December 31, 2011 ·
What about this one? Is it worth $10K? I like this style.

https://www.facebook.com/michael.lloyd.nelson?ref=br_rs#

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 [140]. Web media may change its contents, layout [50, 51], and accessibility [52, 123] on subsequent viewings [105]. Archival Web crawlers like Internet Archive's Heritrix [145] 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 [101], see Appendix A) use Heritrix to preserve the part of the Web relevant to their interest (e.g., content within a specific TLD [69] or about a specific topic) for posterity. Individual archivists may also use this software or other existing Web archiving software to preserve content to a personally owned file store. With individuals' ability to preserve their own

Table 1 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 [49]
public	What I see at `cnn.com`	Large-scale Web crawls (e.g., IA)

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.

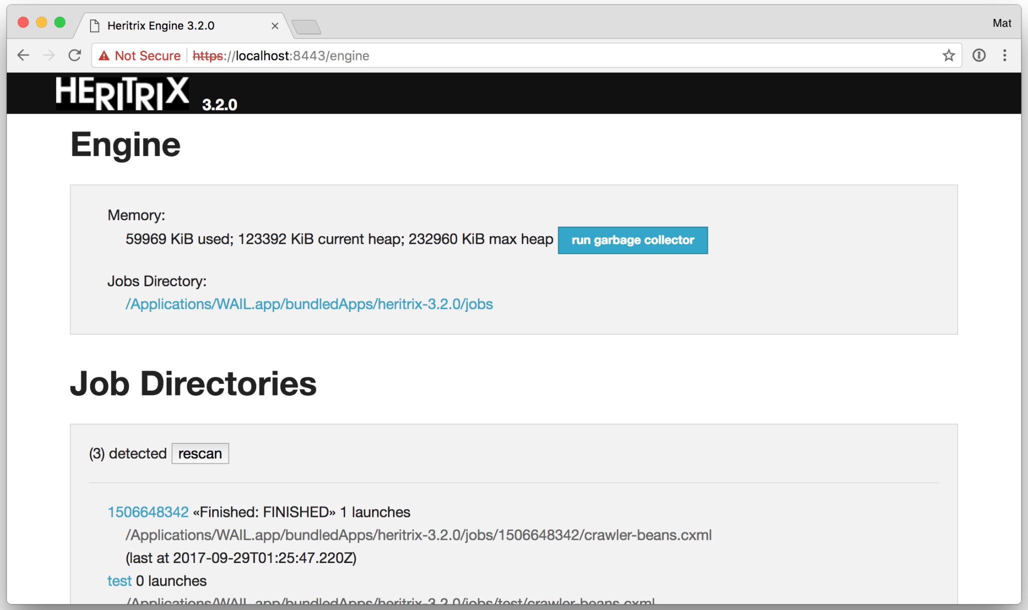
In this work, *personal Web archiving* constitutes Web archiving by an individual without restriction to the availability of content (e.g., private or public) on the live Web. For example, Alice archives what *she* sees on the live Web of a publicly accessible `cnn.com`, her private `facebook.com` news 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 1 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 [26] obtained at crawl time as well as GeoIP-dependent rendering of Web pages [117]. Institutional Web archiving mostly focuses on the publicly available live Web but is not inherently limited to this [25, 84]. Table 2 describes the bounds of some features of Web archives like who administrates 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

Table 2 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

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 [204]. The standardized formats created for Web archiving (e.g., WARC [98]) 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 [32, 125, 168] 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.



Engine

Memory:
59969 KiB used; 123392 KiB current heap; 232960 KiB max heap run garbage collector

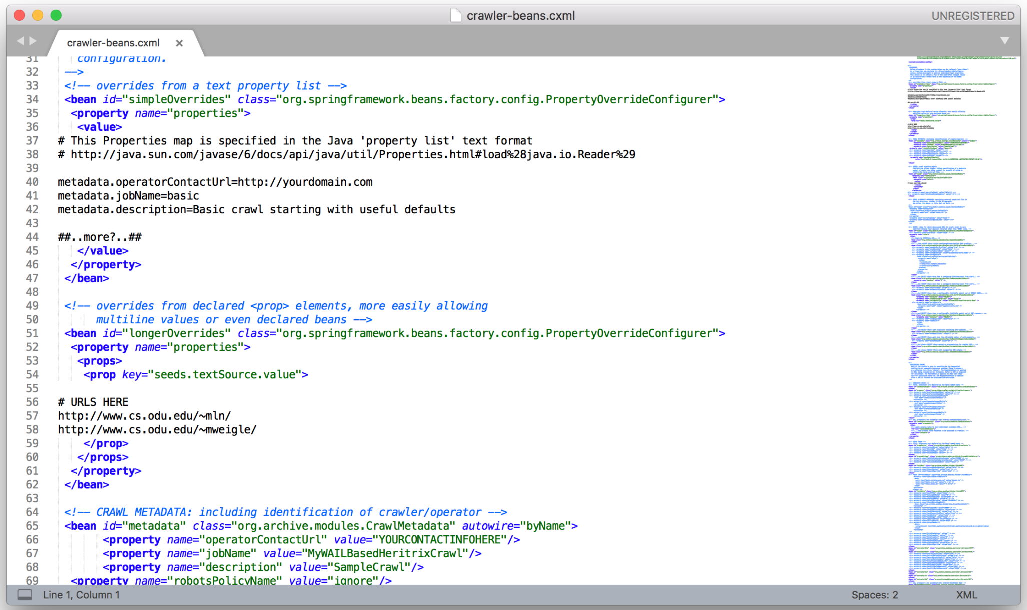
Jobs Directory:
/Applications/WAIL.app/bundledApps/heritrix-3.2.0/jobs

Job Directories

(3) detected rescan

1506648342 «Finished: FINISHED» 1 launches
/Applications/WAIL.app/bundledApps/heritrix-3.2.0/jobs/1506648342/crawler-beans.cxml
(last at 2017-09-29T01:25:47.220Z)
test 0 launches
/Applications/WAIL.app/bundledApps/heritrix-3.2.0/jobs/test/crawler-beans.cxml

(a)



```
   configuration.
 -->
 <!-- overrides from a text property list -->
 <bean id="simpleOverrides" class="org.springframework.beans.factory.config.PropertyOverrideConfigurer">
  <property name="properties">
   <value>
# This Properties map is specified in the Java 'property list' text format
# http://java.sun.com/javase/6/docs/api/java/util/Properties.html#load%28java.io.Reader%29

metadata.operatorContactUrl=http://yourdomain.com
metadata.jobName=basic
metadata.description=Basic crawl starting with useful defaults

##..more?..##
   </value>
  </property>
 </bean>

 <!-- overrides from declared <prop> elements, more easily allowing
      multiline values or even declared beans -->
 <bean id="longerOverrides" class="org.springframework.beans.factory.config.PropertyOverrideConfigurer">
  <property name="properties">
   <props>
    <prop key="seeds.textSource.value">

# URLS HERE
http://www.cs.odu.edu/~mln/
http://www.cs.odu.edu/~mweigle/
    </prop>
   </props>
  </property>
 </bean>

 <!-- CRAWL METADATA: including identification of crawler/operator -->
 <bean id="metadata" class="org.archive.modules.CrawlMetadata" autowire="byName">
      <property name="operatorContactUrl" value="YOURCONTACTINFOHERE"/>
      <property name="jobName" value="MyWAILBasedHeritrixCrawl"/>
      <property name="description" value="SampleCrawl"/>
  <property name="robotsPolicyName" value="ignore"/>
```

(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.

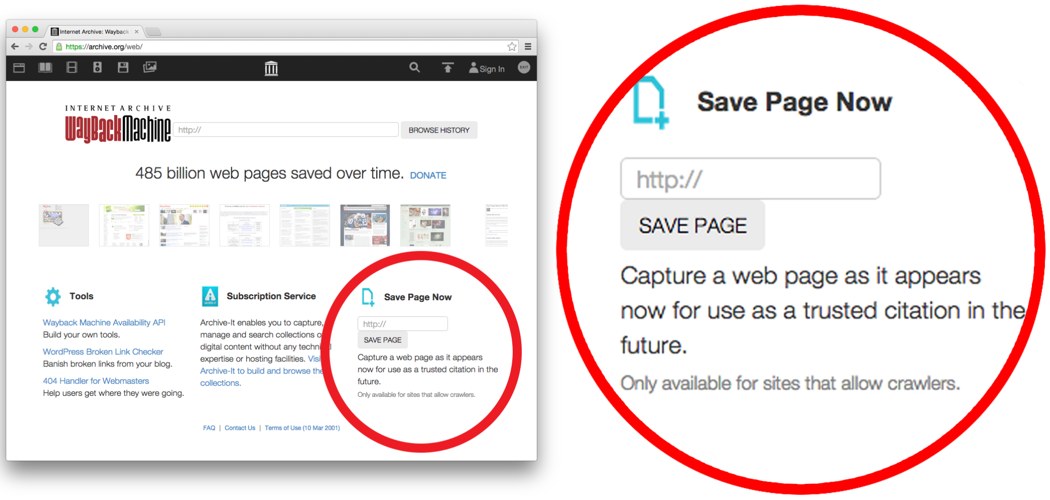
Collaboration and access control are rarely considered in contemporary Web archiving due to the majority of Web archiving efforts targeting publicly-accessible Web content. A reason for this is that the act of preserving Web content by individuals is met with scrutiny of authenticity, i.e., content may have been manipulated prior to capture. Captures performed by those without vetting are not afforded the degree of authenticity as institutional captures [100]. 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 dissertation.

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[3], Imgur[4], and Twitter[5] [143, 162, 161]. 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 pages as conversations are occurring [204].

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



485 billion web pages saved over time.

Wayback Machine Availability API
Build your own tools.
WordPress Broken Link Checker
Banish broken links from your blog.
404 Handler for Webmasters
Help users get where they were going.

Archive-It enables you to capture, manage and search collections o digital content without any techni expertise or hosting facilities. Visi Archive-It to build and browse the collections.

Capture a web page as it appears now for use as a trusted citation in the future.
Only available for sites that allow crawlers.

Save Page Now

http://

SAVE PAGE

Capture a web page as it appears now for use as a trusted citation in the future.

Only available for sites that allow crawlers.

Fig. 4 In 2013 [176], 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 [134]. The Internet Archive[6], an institution set up to preserve content on the public Web [149], has frequently served as a safeguard for those that believed this assumption [141]. 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) [174]. 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[7] (Figure 4) [176]. 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 results in a collection of captures that is quickly outdated as new content is added [134] 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 belongs) for accessing captures of Web content exists in the software implementation of Heritrix and the

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

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

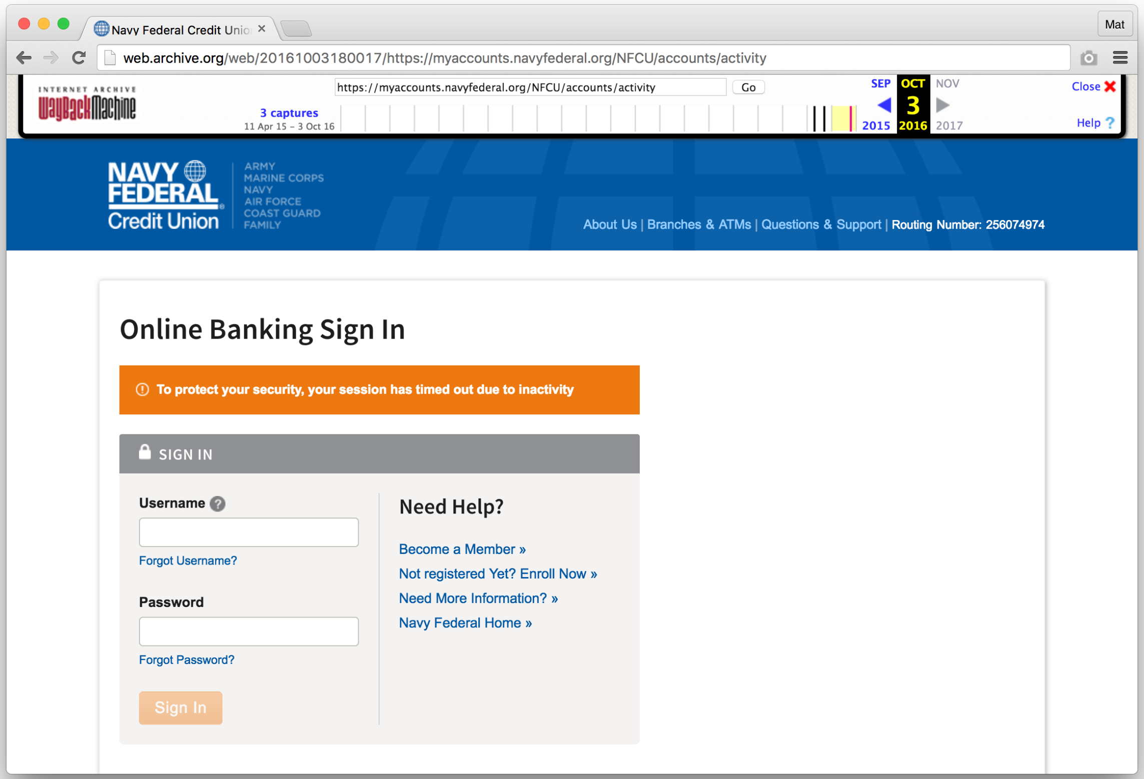
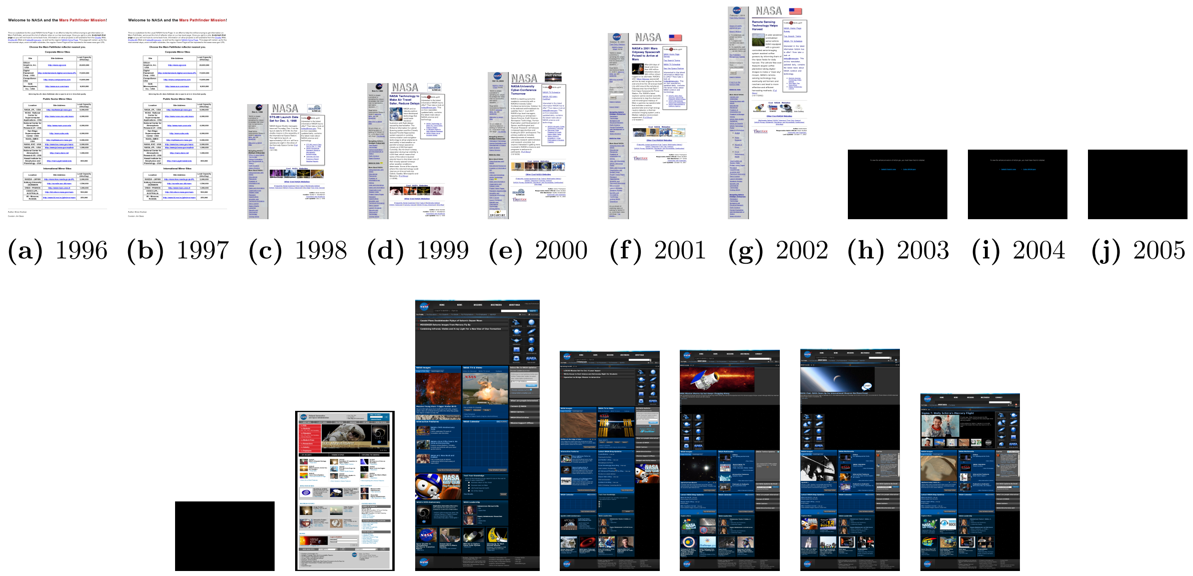

Navy Federal Credit Union
web.archive.org/web/20161003180017/https://myaccounts.navyfederal.org/NFCU/accounts/activity
INTERNET ARCHIVE WaybackMachine
https://myaccounts.navyfederal.org/NFCU/accounts/activity
Go
3 captures
11 Apr 15 – 3 Oct 16
SEP OCT NOV
3
2015 2016 2017
Close
Help
NAVY FEDERAL Credit Union
ARMY MARINE CORPS NAVY AIR FORCE COAST GUARD FAMILY
About Us | Branches & ATMs | Questions & Support | Routing Number: 256074974
Online Banking Sign In
To protect your security, your session has timed out due to inactivity
SIGN IN
Username
Forgot Username?
Password
Forgot Password?
Sign In
Need Help?
Become a Member »
Not registered Yet? Enroll Now »
Need More Information? »
Navy Federal Home »

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.

WARC format [98]. 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.

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) [22]. As an additional caveat, the difference in access mechanism (archival crawler instead of a user's browser) makes it unlikely that the same content that the user wishes to preserve



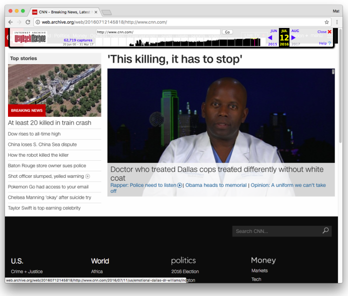
(a) 1996 (b) 1997 (c) 1998 (d) 1999 (e) 2000 (f) 2001 (g) 2002 (h) 2003 (i) 2004 (j) 2005

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

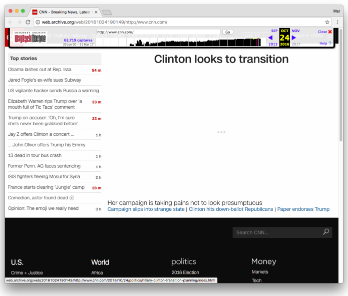
Fig. 6 NASA over time. Changes in design and thus the technologies used is easily observable between 1997 and 1998, 2002 and 2003, 2006 and 2007, and 2007 and 2008. The captures from 2003 to 2006 appear completely black due to the difference in the archival crawler's capability compared to the technology that resided on the page in this time range [118].

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 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` [118] (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 [30]. 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.

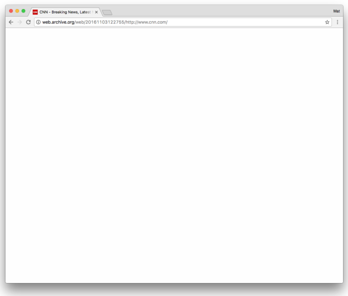
This potential for uncertainty in the reliability of the capture is not limited to content behind authentication. We can see how `cnn.com` looked in September 2016 (Figure 7a),


Top stories
'This killing, it has to stop'
At least 20 killed in train crash
Doctor who treated Dallas cops treated differently without white coat
U.S.
World
politics
Money

(a)


Top stories
Clinton looks to transition
Her campaign is taking pains not to look presumptuous
U.S.
World
politics
Money

(b)



(c)

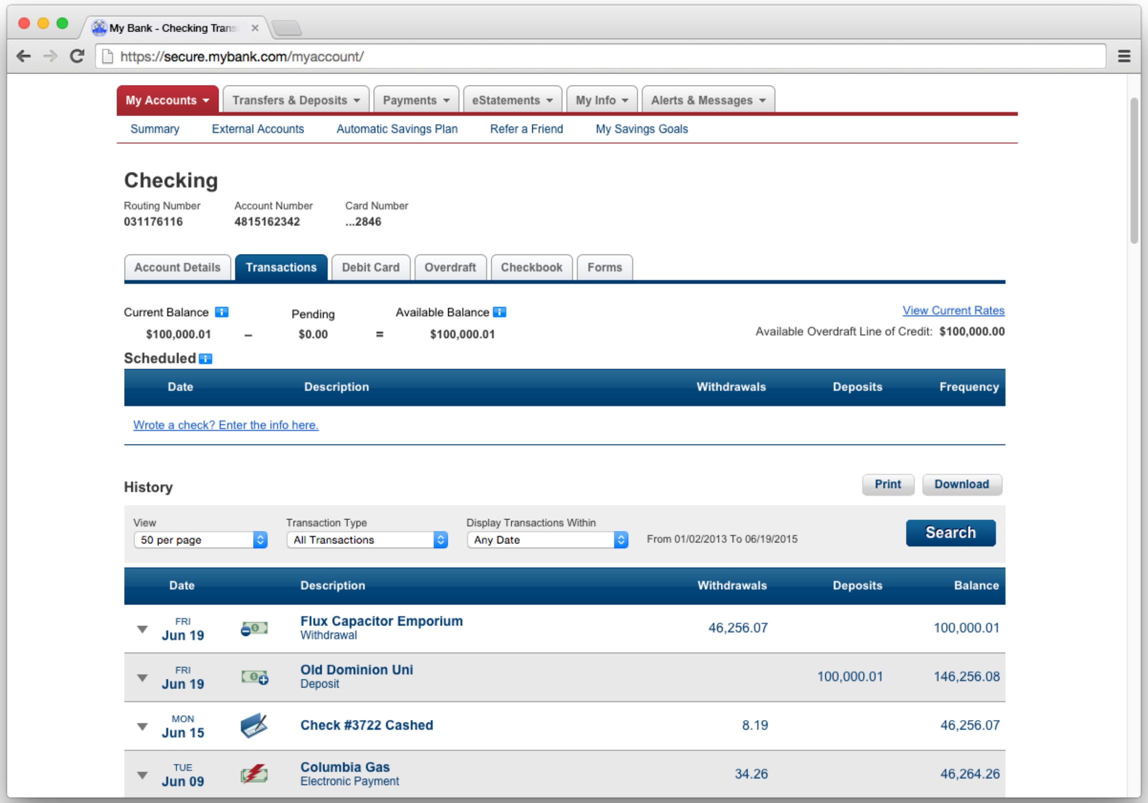
Fig. 7 The quality of the capture over time for three different archived representations for `cnn.com` shows (a) a visually complete[1], (b) visually damaged[2], and (c) very damaged[3] representation.

but some captures are incomplete (Figure 7b) or contain errors that prevent the page from rendering at all (Figure 7c) [30, 31]. 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.

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 [123, 118]. 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 [26]) 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 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,

[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`



My Bank - Checking Trans

https://secure.mybank.com/myaccount/

My Accounts | Transfers & Deposits | Payments | eStatements | My Info | Alerts & Messages

Summary | External Accounts | Automatic Savings Plan | Refer a Friend | My Savings Goals

Checking

Routing Number	Account Number	Card Number
031176116	4815162342	...2846

Account Details | Transactions | Debit Card | Overdraft | Checkbook | Forms

Current Balance		Pending		Available Balance
$100,000.01	–	$0.00	=	$100,000.01

View Current Rates

Available Overdraft Line of Credit: $100,000.00

Scheduled

Date	Description	Withdrawals	Deposits	Frequency

Wrote a check? Enter the info here.

History

Print | Download

View: 50 per page | Transaction Type: All Transactions | Display Transactions Within: Any Date | From 01/02/2013 To 06/19/2015 | Search

Date	Description	Withdrawals	Deposits	Balance
FRI Jun 19	Flux Capacitor Emporium Withdrawal	46,256.07		100,000.01
FRI Jun 19	Old Dominion Uni Deposit		100,000.01	146,256.08
MON Jun 15	Check #3722 Cashed	8.19		46,256.07
TUE Jun 09	Columbia Gas Electronic Payment	34.26		46,264.26

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.

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.

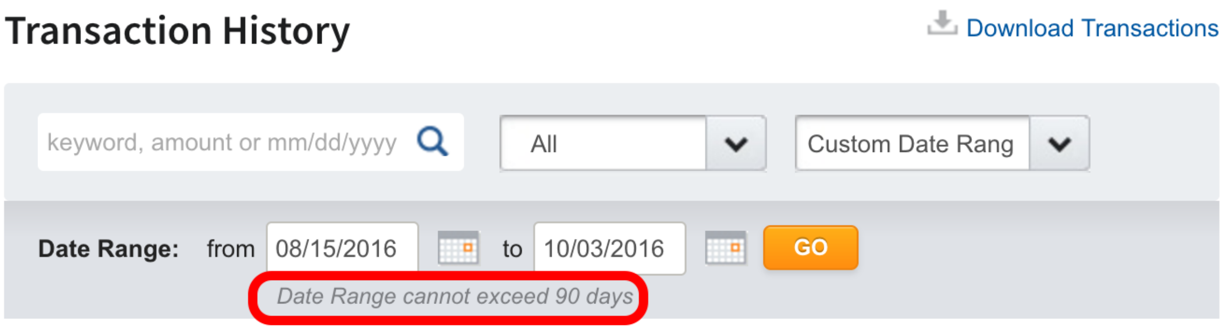
As another example, banks frequently encourage clients to "go paperless", allowing their bank records to be accessed on the bank's Web site (Figure 8) while foregoing paper statements. Oftentimes, banks' sites limit how far in a user's bank history the user may access and how much of the history can 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.

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 [144]. MITRE sought to automatically archive their corporate intranet using Web-scale Web archiving tools [49]. 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 [138].



Transaction History

Download Transactions

keyword, amount or mm/dd/yyyy

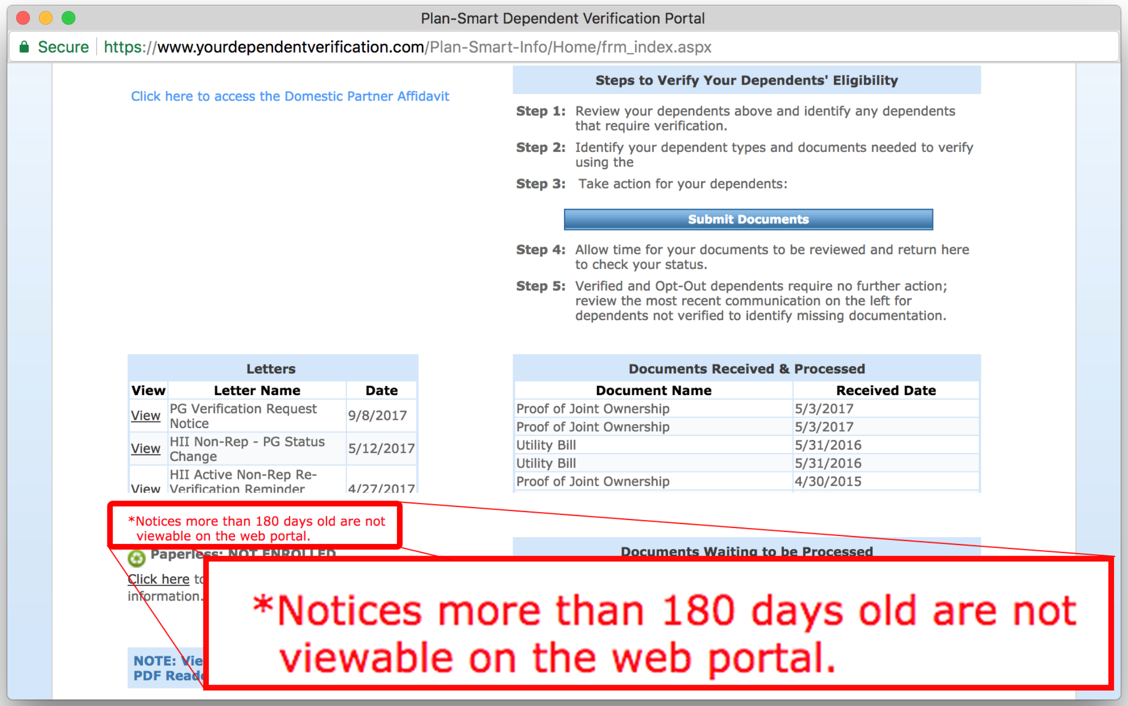
All

Custom Date Rang

Date Range: from 08/15/2016 to 10/03/2016 GO

Date Range cannot exceed 90 days

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



Plan-Smart Dependent Verification Portal

Secure https://www.yourdependentverification.com/Plan-Smart-Info/Home/frm_index.aspx

Click here to access the Domestic Partner Affidavit

Steps to Verify Your Dependents' Eligibility

Step 1: Review your dependents above and identify any dependents that require verification.

Step 2: Identify your dependent types and documents needed to verify using the

Step 3: Take action for your dependents:

Submit Documents

Step 4: Allow time for your documents to be reviewed and return here to check your status.

Step 5: Verified and Opt-Out dependents require no further action; review the most recent communication on the left for dependents not verified to identify missing documentation.

Letters

View	Letter Name	Date
View	PG Verification Request Notice	9/8/2017
View	HII Non-Rep - PG Status Change	5/12/2017
View	HII Active Non-Rep Re-Verification Reminder	4/27/2017

Documents Received & Processed

Document Name	Received Date
Proof of Joint Ownership	5/3/2017
Proof of Joint Ownership	5/3/2017
Utility Bill	5/31/2016
Utility Bill	5/31/2016
Proof of Joint Ownership	4/30/2015

*Notices more than 180 days old are not viewable on the web portal.

Paperless: NOT ENROLLED

Click here to information.

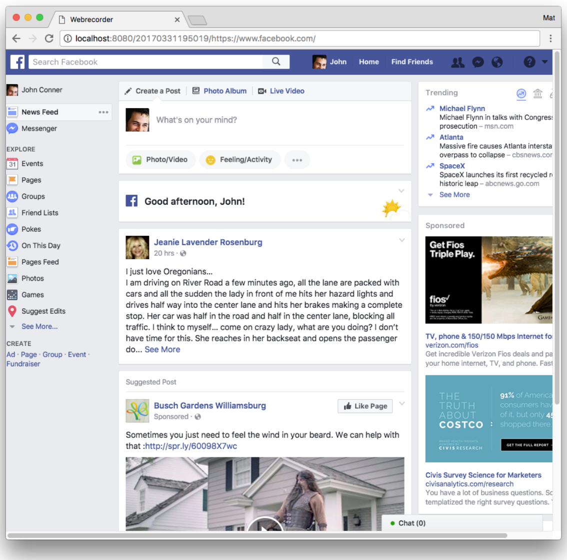
Documents Waiting to be Processed

NOTE: Vie PDF Read

*Notices more than 180 days old are not viewable on the web portal.

(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.



Jeanie Lavender Rosenburg
20 hrs

I just love Oregonians...
I am driving on River Road a few minutes ago, all the lane are packed with cars and all the sudden the lady in front of me hits her hazard lights and drives half way into the center lane and hits her brakes making a complete stop. Her car was half in the road and half in the center lane, blocking all traffic. I think to myself... come on crazy lady, what are you doing? I don't have time for this. She reaches in her backseat and opens the passenger do... See More

Busch Gardens Williamsburg
Sponsored

Sometimes you just need to feel the wind in your beard. We can help with that :http://spr.ly/60098X7wc

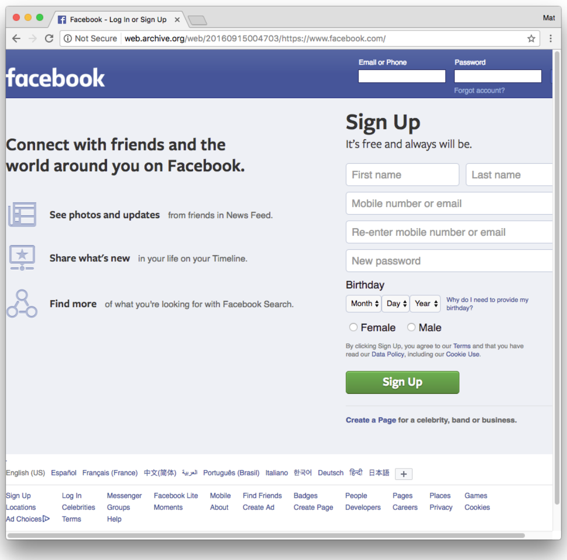
Michael Flynn
Michael Flynn in talks with Congres prosecution – msn.com
Atlanta
Massive fire causes Atlanta intersta overpass to collapse – cbsnews.cor
SpaceX
SpaceX launches its first recycled r historic leap – abcnews.go.com
See More

TV, phone & 150/150 Mbps Internet fo
verizon.com/fios
Get incredible Verizon Fios deals and pa your home internet, TV, and phone. Fast

91% of America consumers hav of it, but only 4 shopped there.

Civis Survey Science for Marketers
civisanalytics.com/research
You have a lot of business questions. Sc templatized the right survey questions.

(a) Local Archive capture



Connect with friends and the world around you on Facebook.

See photos and updates from friends in News Feed.

Share what's new in your life on your Timeline.

Find more of what you're looking for with Facebook Search.

It's free and always will be.

Why do I need to provide my birthday?

By clicking Sign Up, you agree to our Terms and that you have read our Data Policy, including our Cookie Use.

Create a Page for a celebrity, band or business.

(b) Internet Archive capture

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

1.4 RESEARCH QUESTIONS

Our goal is to facilitate the aggregation of public and private Web archives with personal Web archives by mitigating outstanding issues that prevent their aggregation. In this dissertation we define a framework to mitigate outstanding issues relating to private, public, and personal Web archiving. We enumerate multiple outstanding challenges that prevent various types of archives from serving as a more comprehensive picture of how the Web previously existed.

This research provides 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 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?

For the most part, public Web archives capture the public live Web and serve content publicly. The barriers of preservation from the live Web were previously precluded by the technical capability of the tools. In this research we have created tools that allow for preservation of content behind authentication that was previously inaccessible to Web archiving tools. We have also created technical and user-friendly solutions for replaying these captures of potentially private and personal content. We have done this (Chapter 4) in a manner that makes the transitions from the archiving and the end-user replay experience of the live to archived Web more seamless (transition), native (leveraging existing tools and platforms), and familiar (using conventional access paradigms).

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

The accuracy of the historical record as exhibited by Web archives is a function of the capability of the tools to preserve the live Web and replay the captures at a later date. Over time, tools to preserve the live Web have lagged behind in capability relative to Web browsers, the latter of which exhibit the most contemporary features of the Web. We have evaluated the archivability of the tools (Chapter 5) to identify problematic features that have caused the historical record to previously be incomplete.

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

While our previous examinations of archivability as described in the research supporting RQ2 focused on the live Web, the part of the live Web that requires authentication is often not preserved at all. Research in support of this dissertation has mitigated the barriers for preservation of this content, but replaying this content requires a degree of consideration beyond the replay of content that was previously public. We address these issues in the framework introduced in Chapter 7.

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

The expressiveness with regard to syntax and semantics of newfound captures of content behind authentication with the capabilities facilitated by RQ1 is critical for indicating that captures require dynamics for interaction beyond conventional access patterns. Dynamics of interaction differ for users of the archived Web partially due to Web archives largely being agnostic (by design) to the technical features of the archived representations. We provide an extensible solution by adapting recognized Web standards for archival dimension beyond the conventional relation of time (Chapter 6).

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?

Temporally aggregating archived Web representations may require special handling for access to some captures. While these captures may be made accessible by owners of personal and private archives using a mechanism for access control, indicating this necessary dynamic allows for the negotiation procedure to be more systematically exhibited. We provide a mechanism for mitigating the issue of access to captures with this feature through the initial expression that such dynamics are necessary (Chapter 7).

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

Access control standards change with time on the live Web and these standards rarely get propagated to the archived Web. In Chapter 7 we address the need for an extensible framework for access control that facilitates interoperability with the live Web, archived public Web (e.g., CNN.com), public archived Web (e.g., captures at IA), and various permutations of preservation for privacy of the live, public, private, and archived Webs.

1.5 DISSERTATION ROADMAP

Before we can describe our contribution, we first explain the Web, archiving the Web, access control standards as exhibited on the live Web, and interacting with the archived Web in the dimension of time (Chapter 2). Chapter 3 provides details of contemporary research to be considered prior to addressing the research questions answered in this dissertation. Part of this prior research leads to our focus on preservation of the previously neglected part of the live Web. In Chapter 4 we describe our efforts to mitigate this neglect through the creation of accessible tools for the existing or aspiring personal Web archivist.

Previous efforts at preserving the Web have been partially incomplete. In Chapter 5 we describe our studies in evaluating tools and prior efforts at preserving the Web. While these investigation mostly focused on the conventionally preserved public Web, our work seeks to enable the users of the archived Web to be able to interact with Web archives in a manner similar and beyond their interactions with the live Web (Chapter 6).

Re-experiencing the Web of the past with recognition that much of it is personal or private requires consideration when traversing the archived Web in time. In Chapter 7 we provide an in-depth description for a framework for aggregating private and public Web archives. As a framework only has value if useful, we evaluate the design of the framework, perform an evaluation of the implications of integrating the framework, and exhibit the framework through reference implementations in Chapter 8. Finally, Chapter 9 summarizes the conclusions described in this dissertation, our contributions to the state of the art, and future work for potential further exploration in this and related areas.

CHAPTER 2

BACKGROUND

The past is never dead. It's not even past.

- William Faulkner, *Requiem for a Nun [72]*

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 [33] 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. [40] and refined by Fielding et al. [73]. 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 [78, 79, 77, 74, 75, 76] 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 [27]. 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.

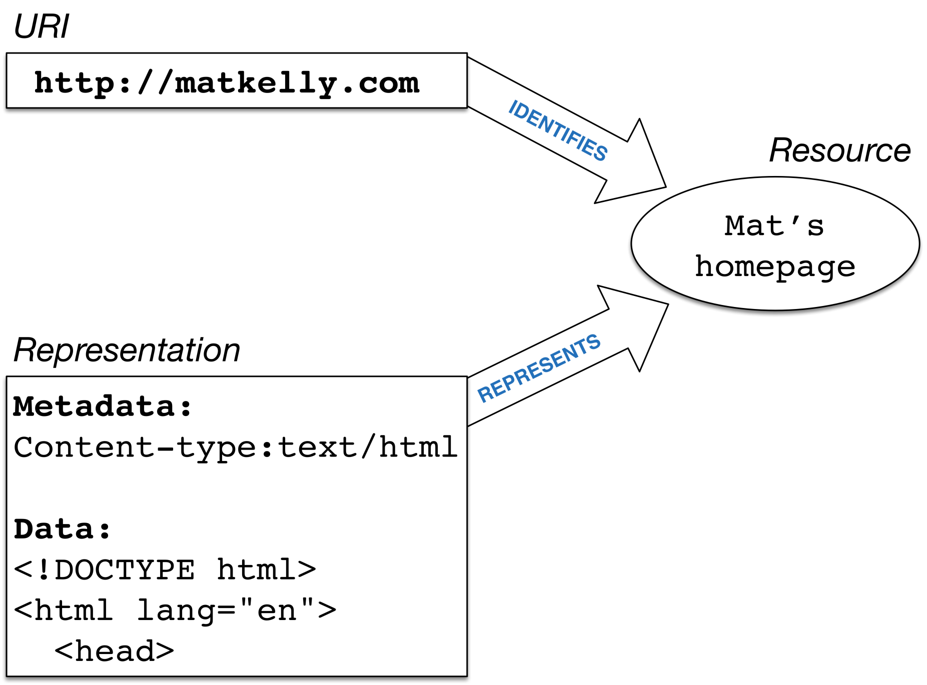


Fig. 11 Sample relation between a URI, resource, and representation on the Web.

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 [39, 34, 132, 37]. 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 [79] 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 [41].

```
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 [97] or a URI [153]. Each link is comma delimited and each link-value is space delimited. The third link shown specifies an extension relation using a URI.

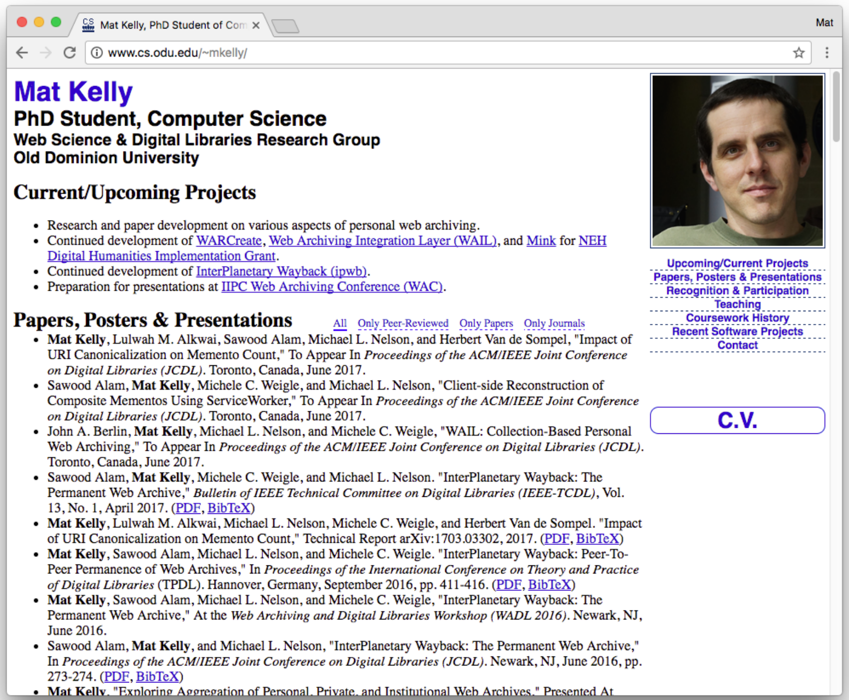
Attributing the relationships between resources on the Web allows expression of resource association. Nottingham (in initial [153] and more recently updated [155] specifications) defined how to represent relations between URIs through use of the `Link` HTTP header.

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 [97] containing the recognized relation types and their semantics, it also allows for "extension relation types". Extension relations are defined with a URI [39] 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 [36, 154, 169]). 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 [35, 182]. 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" [102]). 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.



Mat Kelly
PhD Student, Computer Science
Web Science & Digital Libraries Research Group
Old Dominion University

Current/Upcoming Projects

- Research and paper development on various aspects of personal web archiving.
- Continued development of WARCreate, Web Archiving Integration Layer (WAIL), and Mink for NEH Digital Humanities Implementation Grant.
- Continued development of InterPlanetary Wayback (ipwb).
- Preparation for presentations at IIPC Web Archiving Conference (WAC).

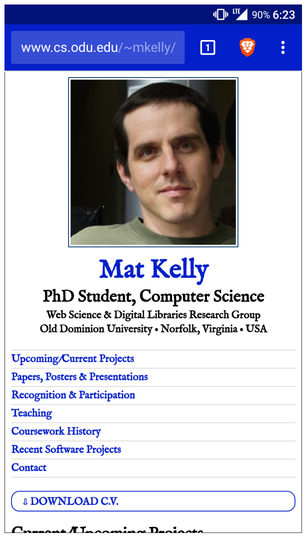
Papers, Posters & Presentations   All Only Peer-Reviewed Only Papers Only Journals

- **Mat Kelly**, Lulwah M. Alkwai, Sawood Alam, Michael L. Nelson, and Herbert Van de Sompel, "Impact of URI Canonicalization on Memento Count," To Appear In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. Toronto, Canada, June 2017.
- Sawood Alam, **Mat Kelly**, Michele C. Weigle, and Michael L. Nelson, "Client-side Reconstruction of Composite Mementos Using ServiceWorker," To Appear In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. Toronto, Canada, June 2017.
- John A. Berlin, **Mat Kelly**, Michael L. Nelson, and Michele C. Weigle, "WAIL: Collection-Based Personal Web Archiving," To Appear In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. Toronto, Canada, June 2017.
- Sawood Alam, **Mat Kelly**, Michele C. Weigle, and Michael L. Nelson. "InterPlanetary Wayback: The Permanent Web Archive," *Bulletin of IEEE Technical Committee on Digital Libraries (IEEE-TCDL)*, Vol. 13, No. 1, April 2017. (PDF, BibTeX)
- **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, 2017. (PDF, BibTeX)
- **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). Hannover, Germany, September 2016, pp. 411-416. (PDF, BibTeX)
- **Mat Kelly**, Sawood Alam, Michael L. Nelson, and Michele C. Weigle, "InterPlanetary Wayback: The Permanent Web Archive," At the *Web Archiving and Digital Libraries Workshop (WADL 2016)*. Newark, NJ, June 2016.
- Sawood Alam, **Mat Kelly**, and Michael L. Nelson, "InterPlanetary Wayback: The Permanent Web Archive," In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. Newark, NJ, June 2016, pp. 273-274. (PDF, BibTeX)
- **Mat Kelly**, "Exploring Aggregation of Personal, Private, and Institutional Web Archives," Presented At

Upcoming/Current Projects
Papers, Posters & Presentations
Recognition & Participation
Teaching
Coursework History
Recent Software Projects
Contact

C.V.

(a)



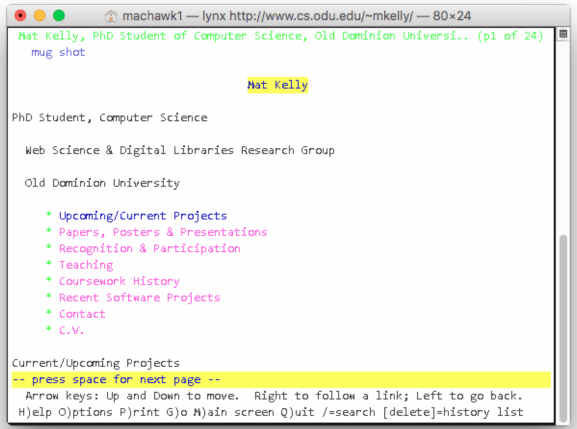
Mat Kelly
PhD Student, Computer Science
Web Science & Digital Libraries Research Group
Old Dominion University • Norfolk, Virginia • USA

Upcoming/Current Projects
Papers, Posters & Presentations
Recognition & Participation
Teaching
Coursework History
Recent Software Projects
Contact

⇩ DOWNLOAD C.V.

Current/Upcoming Projects

(b)



```
 Mat Kelly, PhD Student of Computer Science, Old Dominion Universi.. (p1 of 24)
   mug shot

                                   Mat Kelly

PhD Student, Computer Science

  Web Science & Digital Libraries Research Group

  Old Dominion University

     * Upcoming/Current Projects
     * Papers, Posters & Presentations
     * Recognition & Participation
     * Teaching
     * Coursework History
     * Recent Software Projects
     * Contact
     * C.V.

Current/Upcoming Projects
-- press space for next page --
  Arrow keys: Up and Down to move.  Right to follow a link; Left to go back.
 H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

(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 [44]), and (c) a desktop terminal-based Web browser (Lynx [64]) 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 [64]), but the Web is also accessible with command-line or scripting tools, as with the "curl" user-agent in Figure 15 [38]. 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 [88], 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 [153] 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 [90] (e.g., the style.css file in Figure 12), relational resources in HTTP need not be format-specific [153].

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) [80]. Many archival Web crawlers (e.g., Internet Archive's Heritrix [145]) 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. This process is performed recursively (e.g., more URIs are "discovered" when subsequent scripts are executed) in an attempt to dynamically and adaptably acquire the "deferred" representations [48] 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 [73] 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 [79]. 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 [79]. 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 [188] 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 [79] but, as stated by Snell, the requirements 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 dissertation. In Section 7.2.1 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 [26] standardized the specification of HTTP State Management via Cookies, as was previously defined by Kristol and Montulli in two preceding specifications [130, 131]. 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 6.1.

Holtman and Mutz [91] 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 [156] 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.

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

```
$ 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.

is also as a container of the content, which allows it to be interpreted in a variety of ways [140] (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 [25] in 2016 (and previously in 2011 [148] and 2013 [24]) 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" [25]. This is problematic, as we stated with our bearer examples in Chapter 1.

In 2011, Gomes et al. [84] 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 in 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 [158] enumerating 17 major initiatives at the time and another study performed by the Joint Information Systems Committee (JISC) [60] 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 not wish to disclose their procedure and holdings for reasons that we hope to mitigate.

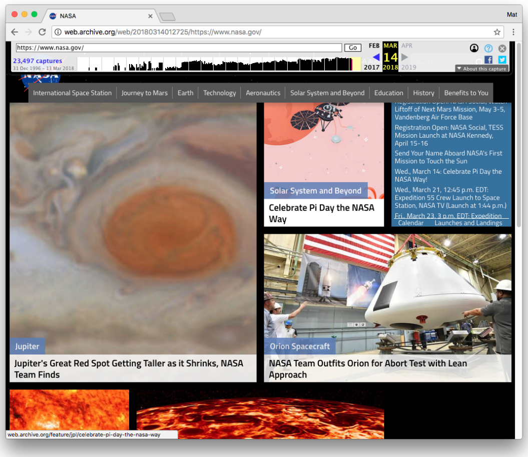
Other Web archives exist beyond Internet Archive. Some Web archives like the UK Web Archive (Appendix A) are scoped to only preserve certain parts of the Web – in this case, only "UK Web sites". Other Web archives like archive.is [150] and WebCite [71] (Appendix A) allow submission of Web pages to be 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.

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 [185]. Heritrix [145] 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 [168] and WARCreate [125]. 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

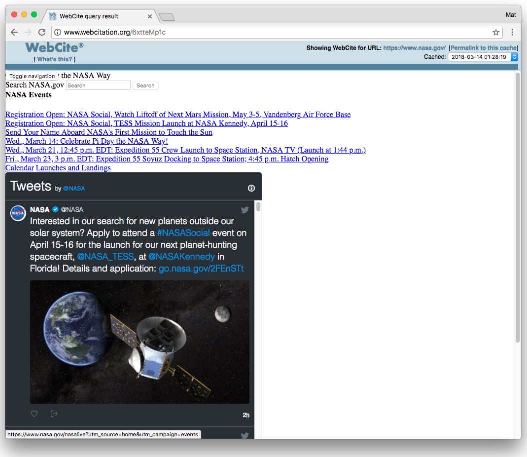


Liftoff of Next Mars Mission, May 3-5, Vandenberg Air Force Base
Registration Open: NASA Social, TESS Mission Launch at NASA Kennedy, April 15-16
Send Your Name Aboard NASA's First Mission to Touch the Sun
Wed., March 14: Celebrate Pi Day the NASA Way!
Wed., March 21, 12:45 p.m. EDT: Expedition 55 Crew Launch to Space Station, NASA TV (Launch at 1:44 p.m.)
Fri., March 23, 3 p.m. EDT: Expedition

Celebrate Pi Day the NASA Way

Jupiter's Great Red Spot Getting Taller as it Shrinks, NASA Team Finds

NASA Team Outfits Orion for Abort Test with Lean Approach

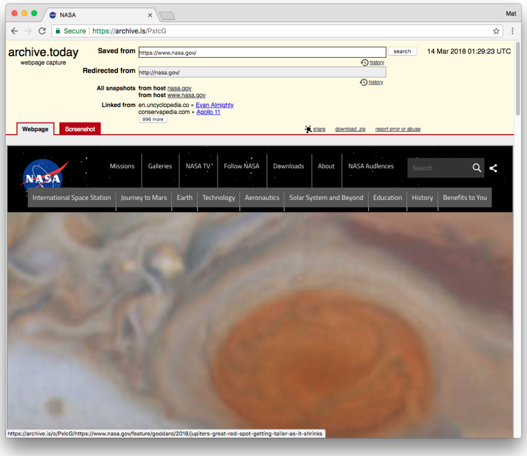

WebCite query result
Mat
www.webcitation.org/6xtteMp1c
WebCite®
[What's this?]
Showing WebCite for URL: https://www.nasa.gov/ [Permalink to this cache]
Cached: 2018-03-14 01:28:19
Toggle navigation
Search
Search
Tweets by @NASA
NASA @NASA
2h
https://www.nasa.gov/nasalive?utm_source=home&utm_campaign=events

the NASA Way
Search NASA.gov
NASA Events

Registration Open: NASA Social, Watch Liftoff of Next Mars Mission, May 3-5, Vandenberg Air Force Base
Registration Open: NASA Social, TESS Mission Launch at NASA Kennedy, April 15-16
Send Your Name Aboard NASA's First Mission to Touch the Sun
Wed., March 14: Celebrate Pi Day the NASA Way!
Wed., March 21, 12:45 p.m. EDT: Expedition 55 Crew Launch to Space Station, NASA TV (Launch at 1:44 p.m.)
Fri., March 23, 3 p.m. EDT: Expedition 55 Soyuz Docking to Space Station; 4:45 p.m. Hatch Opening
Calendar Launches and Landings

Interested in our search for new planets outside our solar system? Apply to attend a #NASASocial event on April 15-16 for the launch for our next planet-hunting spacecraft, @NASA_TESS, at @NASAKennedy in Florida! Details and application: go.nasa.gov/2FEnSTt

(a) Internet Archive (b) WebCite



Saved from https://www.nasa.gov/ 14 Mar 2018 01:29:23 UTC
Redirected from http://nasa.gov/
All snapshots from host nasa.gov
from host www.nasa.gov
Linked from en.uncyclopedia.co » Evan Almighty
conservapedia.com » Apollo 11

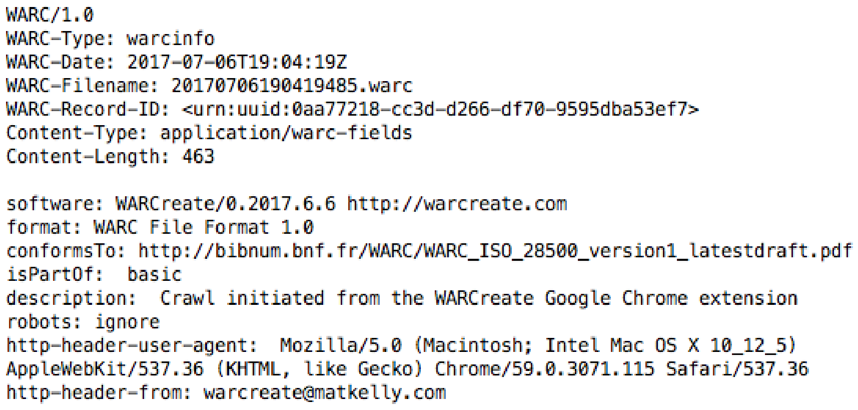
(c) Archive.is

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.

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.

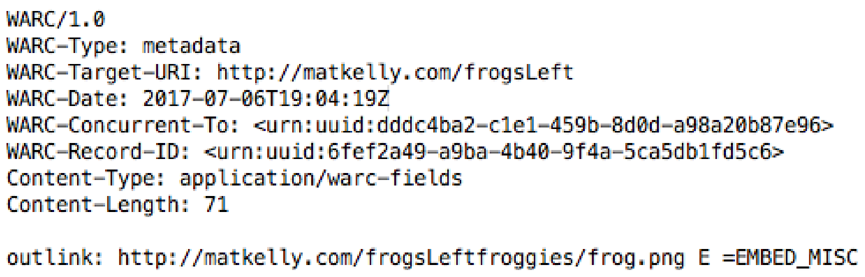
2.4.3 THE WEB ARCHIVE (WARC) FORMAT

Captures of the live Web by Heritrix and many other tools are often stored using the standard Web ARChive (WARC) format [98]. 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 [98], 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 [186] 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 [81]). 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.



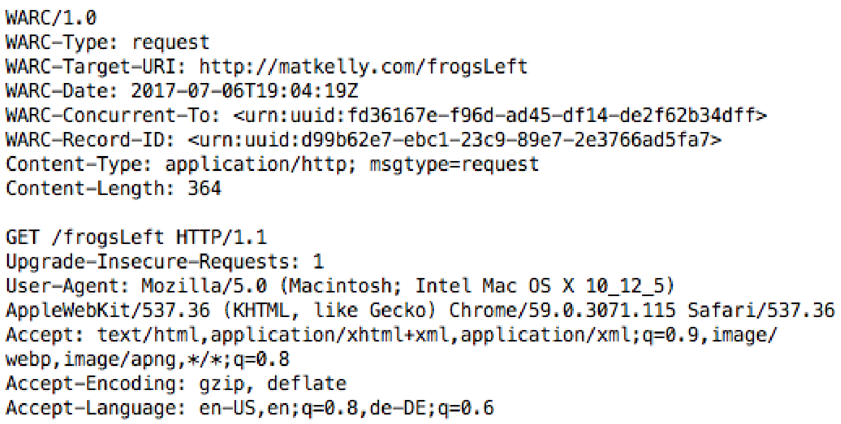
```
WARC/1.0
WARC-Type: warcinfo
WARC-Date: 2017-07-06T19:04:19Z
WARC-Filename: 20170706190419485.warc
WARC-Record-ID: <urn:uuid:0aa77218-cc3d-d266-df70-9595dba53ef7>
Content-Type: application/warc-fields
Content-Length: 463

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



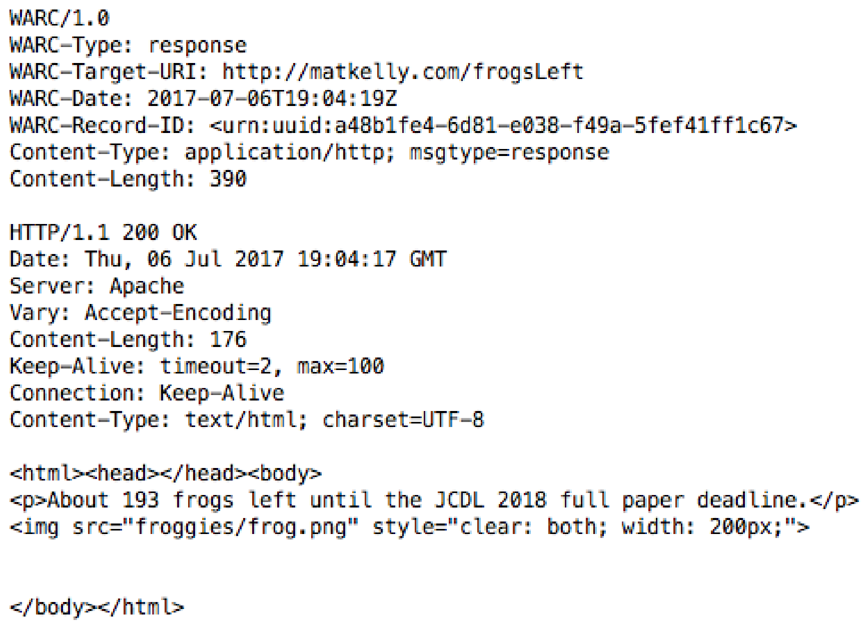
```
WARC/1.0
WARC-Type: metadata
WARC-Target-URI: http://matkelly.com/frogsLeft
WARC-Date: 2017-07-06T19:04:19Z
WARC-Concurrent-To: <urn:uuid:dddc4ba2-c1e1-459b-8d0d-a98a20b87e96>
WARC-Record-ID: <urn:uuid:6fef2a49-a9ba-4b40-9f4a-5ca5db1fd5c6>
Content-Type: application/warc-fields
Content-Length: 71

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

(b) `metadata` record



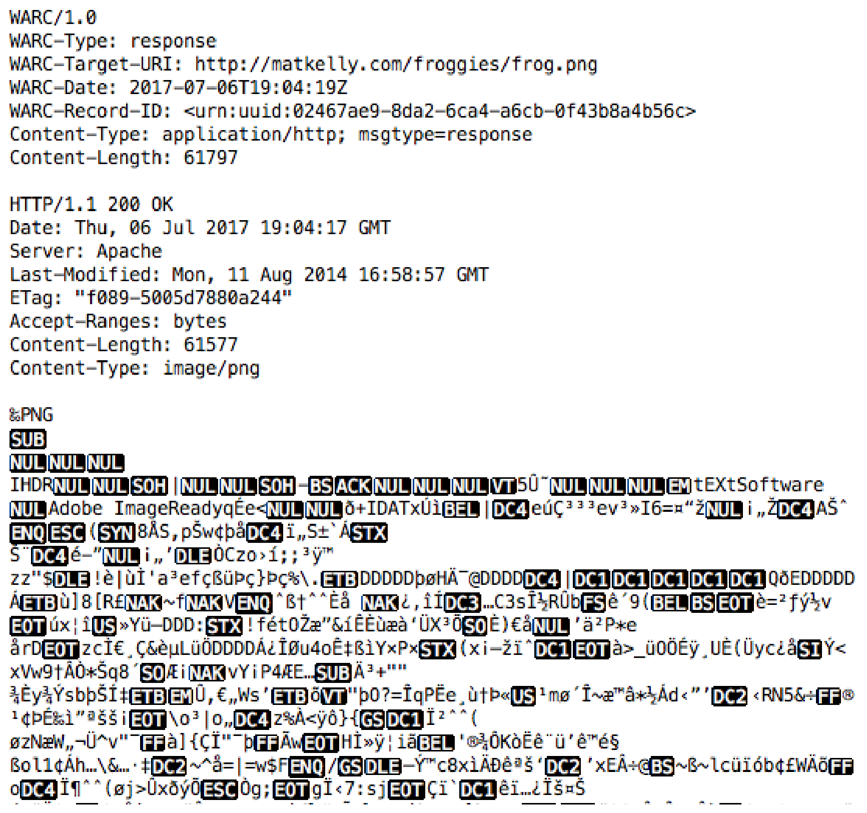
```
WARC/1.0
WARC-Type: request
WARC-Target-URI: http://matkelly.com/frogsLeft
WARC-Date: 2017-07-06T19:04:19Z
WARC-Concurrent-To: <urn:uuid:fd36167e-f96d-ad45-df14-de2f62b34dff>
WARC-Record-ID: <urn:uuid:d99b62e7-ebc1-23c9-89e7-2e3766ad5fa7>
Content-Type: application/http; msgtype=request
Content-Length: 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
NULNULNUL
IHDRNULNULSOH|NULNULSOH–BSACKNULNULNULVT5Û˜NULNULNULEMtEXtSoftware
NULAdobe ImageReadyqÉe<NULNULð+IDATxÚìBEL|DC4eúÇ³³³ev³»I6=¤“žNULi„ŽDC4AŠˆ
ENQESC(SYN8ÅS,pŠw¢þåDC4ï„S±`ÁSTX
Š¨DC4é–"NULi„'DLEÒCzo›í;;³ÿ™
zz"$DLE!è|ùÌ'a³efçßüÞç}Þç%\.ETBDDDDDþøHÄ¯@DDDDDC4|DC1DC1DC1DC1DC1QðEDDDDD
ÁETBù]8[R£NAK~fNAKVENQˆßt^^Èå NAK¿,îÏDC3…C3sÎ½RÛbFSê´9(BELBSEOTè=²fý½v
EOTúx¦îUS»Yü—DDD:STX!fétOŽæ"&íÊÈùæà'ÜX³ÕSOÈ)€åNUL'ä²P*e
årDEOTzcÌ€¸Ç&èµLüÖDDDDÁ¿Îøu4oÊ‡ßìY×P×STX(xi–žïˆDC1EOTà>_üOÕÉÿ¸UÈ(Üyc¿åSIÝ<
xVw9†ÃÒ*Šq8´SOÆiNAKvYiP4ÆE…SUBÄ³+""
¾Ėy¾Ýsbþ$Í‡ETBEMÛ,€„Ws'ETBõVT"þ0?=ÎqPËe¸ù†Þ«US¹mø´Î~æ™â*½Ád‹"'DC2‹RN5&÷FF®
¹¢ÞÉ‰ì"ªšši EOT\o³|o„DC4z%À<ÿô}{GSDC1Ï²^^(
øzNæW„¬Ü^v"¯FFà]{ÇÏ"¯þFFÃwEOTHÌ»ÿ¦iãBEL'®¾ÔKòËê¨ü'ê™é§
ßol1¢Áh…\&…·‡DC2~^å=|=w$FENQ/GSDLE–Ÿ™c8xìÄÐêªš'DC2'xEÂ÷@BS~ß~lcüïóbC£WÄõFF
oDC4Ï¶^^(øj>ÜxðýÕESCÒg;EOTgÏ‹7:sjEOTÇi`DC1êï…¿Ìš¤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 [99] 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) [96] 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 [116, 115]. 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` 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 are not 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 files of WARC's predecessor, the ARC format [55], 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 [47]), the contents requested for a URI or an embedded resource on a page may exist in multiple WARC files. Internet Archive's *Wayback Machine* [193], its open-source derivative project *OpenWayback* [94], and *pywb* [129] 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 [198] 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).

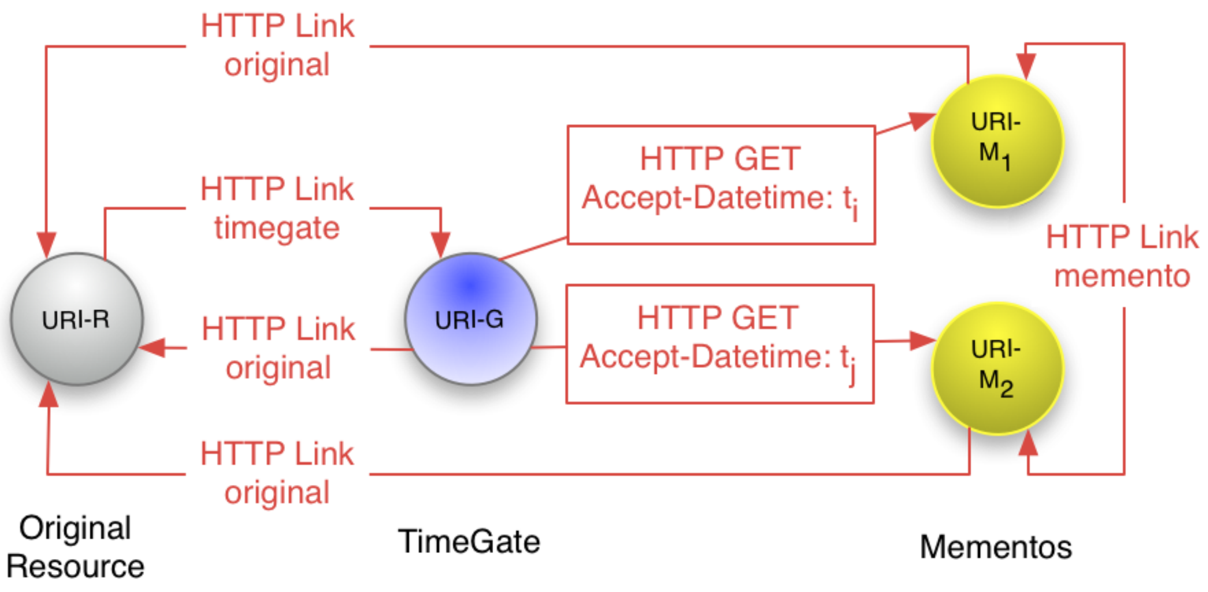


Fig. 21 Memento provides the ability to associate live Web and archived Web captures (at URI-Rs and URI-Ms, respectively), relations between URI-Ms for the same URI-R, and negotiation of resolving a datetime closest to one specified in an HTTP Accept-Datetime header using a TimeGate (at URI-G) [2].

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` [19]), 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 [43] 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 [79]. 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 capture 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"
...
```

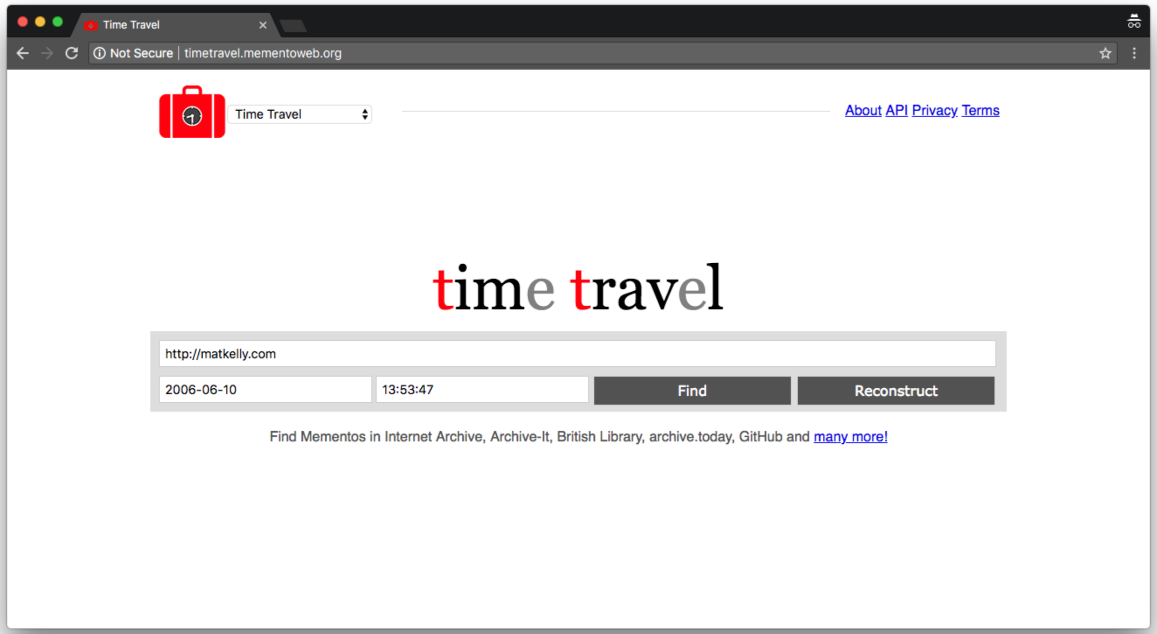
Fig. 23 When a user-agent receives a redirect (Figure 22) when dereferencing the URI-G, the URI-M returned from the TimeGate is subsequently requested and the HTTP response returned to the user-agent for the user.

```
$ 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 [153] 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 [45] and CDXJ [6, 11], 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.

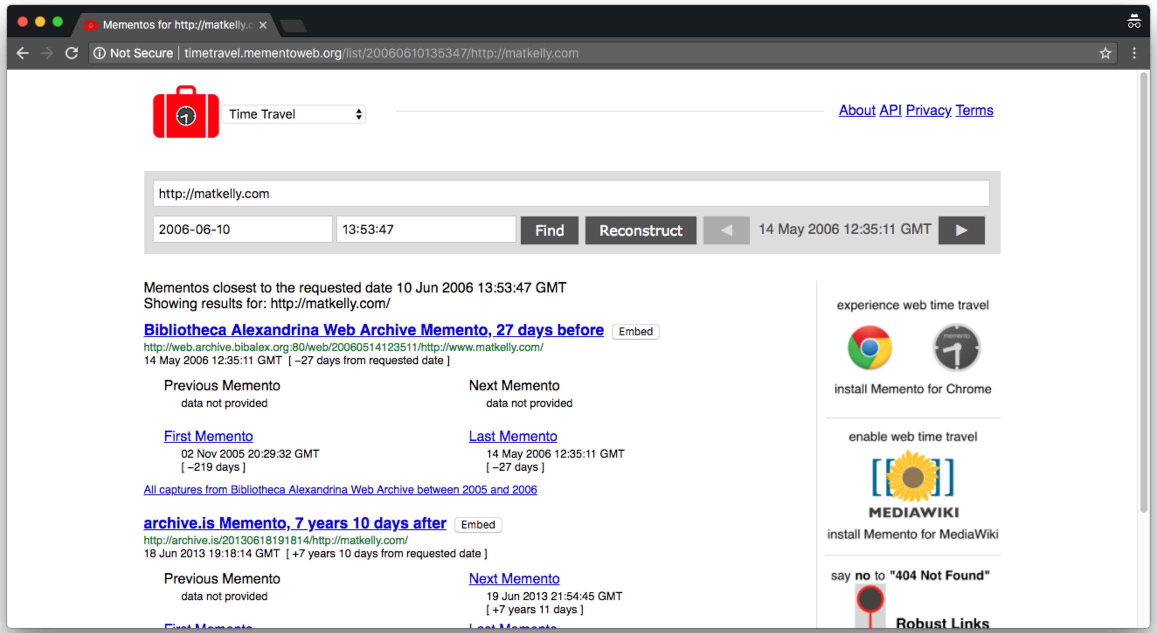


http://matkelly.com

2006-06-10 13:53:47

Find Mementos in Internet Archive, Archive-It, British Library, archive.today, GitHub and many more!

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


Mementos for http://matkelly.c
Not Secure | timetravel.mementoweb.org/list/20060610135347/http://matkelly.com
Time Travel
About API Privacy Terms
Find
Reconstruct
experience web time travel
install Memento for Chrome
enable web time travel
MEDIAWIKI
install Memento for MediaWiki
say no to "404 Not Found"
Robust Links

http://matkelly.com

2006-06-10 13:53:47 14 May 2006 12:35:11 GMT

Mementos closest to the requested date 10 Jun 2006 13:53:47 GMT
Showing results for: http://matkelly.com/

Bibliotheca Alexandrina Web Archive Memento, 27 days before Embed
http://web.archive.bibalex.org:80/web/20060514123511/http://www.matkelly.com/
14 May 2006 12:35:11 GMT [–27 days from requested date]

Previous Memento
data not provided

Next Memento
data not provided

First Memento
02 Nov 2005 20:29:32 GMT
[–219 days]

Last Memento
14 May 2006 12:35:11 GMT
[–27 days]

All captures from Bibliotheca Alexandrina Web Archive between 2005 and 2006

archive.is Memento, 7 years 10 days after Embed
http://archive.is/20130618191814/http://matkelly.com/
18 Jun 2013 19:18:14 GMT [+7 years 10 days from requested date]

Previous Memento
data not provided

Next Memento
19 Jun 2013 21:54:45 GMT
[+7 years 11 days]

(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 [153] 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 later applied to the archived Web.

OAuth 2.0 [86] 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 [103]. 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.

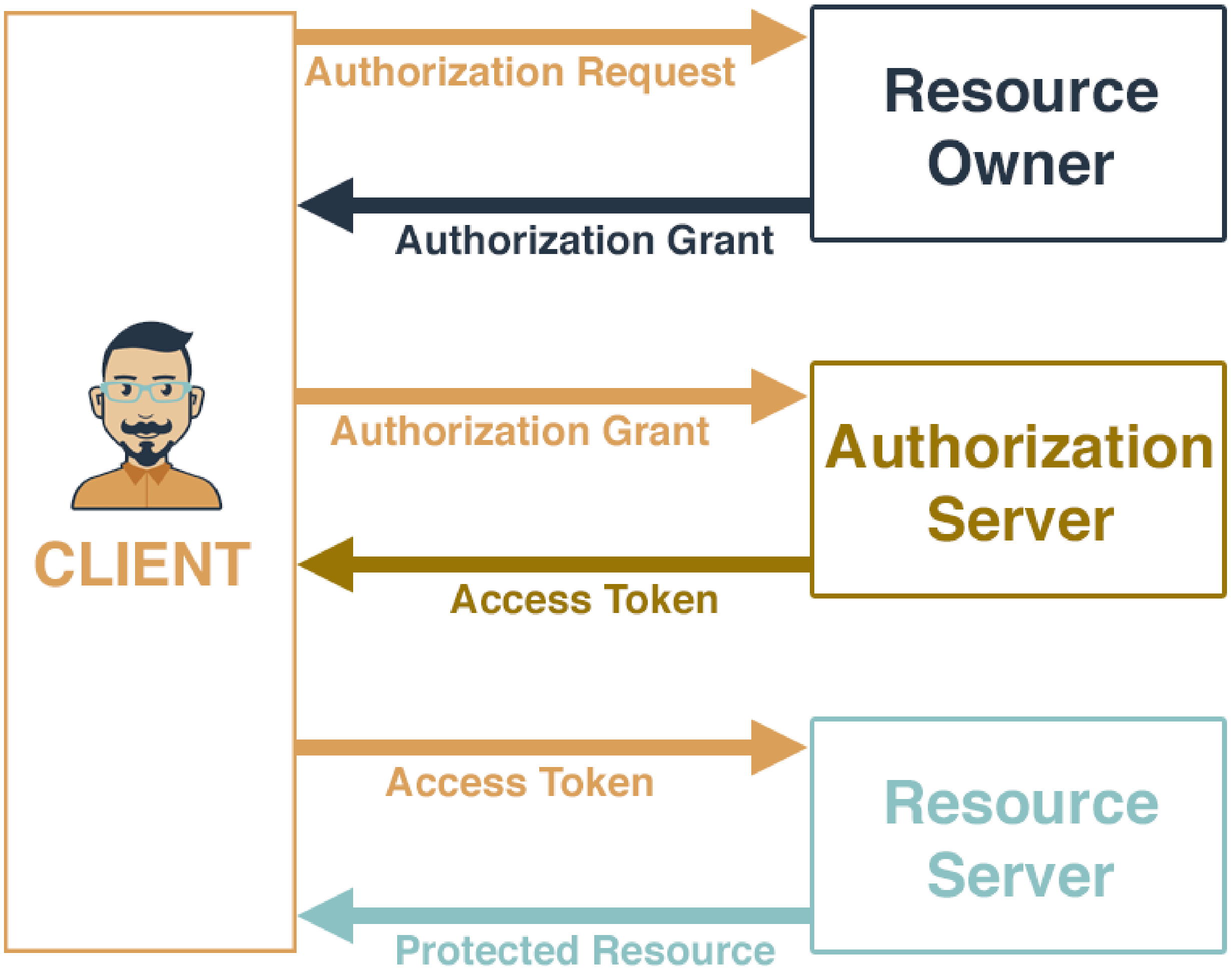


Fig. 27 The OAuth 2.0 abstract protocol flow decouples the resource owner, resource server, and authorization client using a token-based system for access persistence.

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 persistence prevents the user from 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

...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 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 section will address the aggregation aspects between these three archive types, 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 [116, 115], we performed a deep dive into the identifier for mementos (URI-Ms), highlighting the limitations 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 themselves 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. In Section 6.2.1 we build upon the methodologies of these preliminary works by abstracting what we defined as "content-based attributes" that would have mitigated the issues described. We extrapolate this abstraction onto other attribute types (e.g., derived attributes that require further calculation in Section 6.2.2) for expression in TimeMaps.

3.1.2 MEMENTO AGGREGATION

Memento currently allows URIs for mementos, TimeMaps, and TimeGates (URI-Ms, URI-Ts, and URI-Gs, respectively) to be aggregated (à la Section 2.5) by a Memento aggregator and returned to a user sending requests to a Memento endpoint. The 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 [10] 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 (Section 7.1.1) to account for the privacy and access control aspects beyond the considerations that the Memento framework addresses.

Rosenthal et al. [175] described the usage of Memento aggregation for content with restricted access. A library using their LOCKSS system could see their own captures inline with other archives provided by a fallback aggregator like the Time Travel service. The concepts of listing mementos with restricted access, hierarchical relations between aggregators, and query precedence and short-circuiting are discussed in Sections 7.1.2, 7.1.1, and 7.2.2, respectively.

AlSum et al. [18, 17] 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 [183]. 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. We explore client-assisted Memento aggregation using the `Prefer` header [114] to mitigate some of the downsides of the broadcasting approach, detailed in Section 6.1.

Brunelle and Nelson [53] 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 dissertation, as Web archives hosted by individuals are likely to be less consistent with uptime and reliability compared to their institutional Web archive counterparts. In Section 4.5 we describe our efforts toward the persistence of personal Web archives using a collaborative and secure approach at propagating personal Web archives' holdings.

Bornand et al. [42] 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). In Section 7.1.1, we extend the functionality of conventional Memento aggregators through the introduction of a Memento Meta-Aggregator. This additional abstraction allows for a more systematic means for users to control the archives that are queried by the aggregator (Section 6.1), a capability beyond the scope of a conventional aggregator's static set of archives, as considered by Bornand et al.

Alam et al. [12, 11] 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 described in Section 7.3.5. However, because a personal or private Web archive's holdings are likely much smaller than most institutions, the rate of unnecessary requests to personal archives when aggregated would likely be much higher. The work of Alam et al. helped to inform the design of the modified aggregator in this dissertation (evaluated in Section 8.1) 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. Expressing this from the perspective of aggregating private captures as a subset of all sources is addressed in Chapter 8 by providing a means of preventing a private Web archive's holdings from being unnecessarily exposed.

In 2013, Rosenthal [172] 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 dissertation (RQ4 and RQ5) 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 (Section 6.2). Rosenthal [171] 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 [170] 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 [170] 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 (as facilitated by short-circuiting in Section 7.2.2) from the results and redirecting a user to a version of the TimeMap containing only captures that are not a login page (as discussed in Section 7.3.6).

3.1.3 TYPES OF MEMENTOS

Jones et al. [104] discussed obtaining the "raw mementos" consisting of un-rewritten links without archival banners in captures in a systematic way using the HTTP Link response header. By utilizing the HTTP Prefer request header [188] (previously discussed in

Section 2.3), 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. We leverage `Prefer` extensively in providing the ability for negotiating with Web archives in dimensions beyond time (Chapter 6).

Van de Sompel et al. [197] 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 [173] 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. We [8] provided an alternate approach to mitigating the archived representation rewriting problem using a client-side rerouting mechanism through the use of Service Workers.

The work in this dissertation 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 (using StarMaps, described in Chapter 6) to mitigate issues that occur before and while dereferencing URI-Ms. A goal of this work is to further involve the client in the aggregation process (e.g., client-side archival specification in Section 6.1). Interaction with the aggregators through these sort of mechanisms will be a first step in accomplishing the goals of the framework.

In previous work [108], 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 [117] extended this work by identifying personalized representations of mementos and providing a mechanism to navigate between additional dimensions beyond time. We explore this much further in Chapter 6. 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 is useful to distinguish URI-Ms that represent personalized mementos, mementos that were originally behind authentication (using attributes discussed in Section 6.2.3), and mementos in personal Web archives that require additional considerations and mechanisms to access (RQ4 and RQ5).

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, exposing that captures exists for a URI-R in a private Web archive (even without necessarily exposing the contents) might encourage those trying to illegitimately access the private capture to proceed. With one objective of this work being to facilitate aggregation of these captures (Chapter 7), 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 [139] 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." We enable individuals to preserve this content themselves (Section 4.2) to mitigate the problems and barriers of institutions preserving it. For individuals to preserve this content was originally beyond the scope of Marshall and Shipman due to the technical limitations of archiving tools at the time and the fact that Web archiving was for the most part limited to institutional efforts.

Lindley et al. [134] 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.” We previously briefly highlighted a use case relating to Flickr preservation and privacy in Chapter 1. We evaluate the impact of facilitating this disassociation of a single user to captures as exhibited in the Chapter 1 scenarios in Section 8.4. We also provide scenarios where this decoupling of personas can be re-associated from the user end in Section 7.3.5.

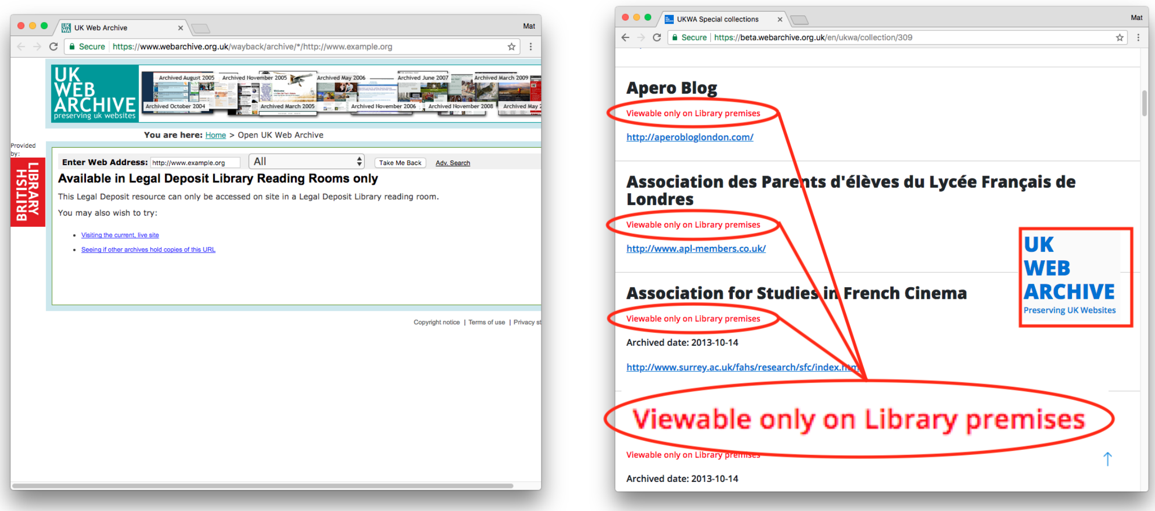
3.2.2 ACCESS CONTROL IN WEB ARCHIVE PRACTICE

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 [46]. Accessing this same capture while on-site at the archive permits access. In early 2018, the UKWA began migrating [203] to using an adapted version [195] 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.



https://www.webarchive.org.uk/wayback/archive/*/http://www.example.org

You are here: Home > Open UK Web Archive

Enter Web Address: http://www.example.org

Available in Legal Deposit Library Reading Rooms only

This Legal Deposit resource can only be accessed on site in a Legal Deposit Library reading room.

You may also wish to try:

- Visiting the current, live site
- Seeing if other archives hold copies of this URL

https://beta.webarchive.org.uk/en/ukwa/collection/309

Apero Blog

Viewable only on Library premises

http://aperobloglondon.com/

Association des Parents d'élèves du Lycée Français de Londres

Viewable only on Library premises

http://www.apl-members.co.uk/

Association for Studies in French Cinema

Viewable only on Library premises

Archived date: 2013-10-14

http://www.surrey.ac.uk/fahs/research/sfc/index.htm

Viewable only on Library premises

Viewable only on Library premises

Archived date: 2013-10-14

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 professor 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 [189] 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 Section 4.5, we facilitate a more decentralized and distributed 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 IP address and is implemented via Apache's .htaccess file [160]. 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 [152] 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 dynamics are explored in Section 7.1.2 in the context of private Web archives. The introduction of additional attributes relating to access (Section 6.2.3) helps to mitigate URI collisions, as occur with captures at identical URI-Rs with different levels of access (Section 1.2). iProxy provided users a means of archiving and replay with access parameters that extended URLs with commands for retrieval [165]. Because of the additional attributes beyond public-private in archives (Section 6.2), simple URI 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

using this means is insufficient for lookup in private Web archives whose content was behind authentication on the live Web. We address this in Chapter 6.

3.4 ARCHIVING: PUBLIC VS. PRIVATE VS. PERSONAL

In Chapter 7 we describe a framework for aggregating public and private Web archives. To aggregate these different classes of archives, it helps to first understand what each class 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 focusing on each class.

3.4.1 PUBLIC WEB ARCHIVING

Brunelle [48, 54] 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 extend on this work in Section 4.3.2 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 (Section 4.2).

Access to Web archives is a fundamental theme in this dissertation. Ben-David and Huurdeman [28] 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. [92]) 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 conventional means of archival access is by URI, users will likely wish to access their private, personal, and even institutional public Web captures by alternate means. This was beyond the scope of Ben-David and Huurdeman, who focused on public Web archives.

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 [58] 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. In Chapter 6 we discuss details on access beyond URI and time using an extensible approach to build on dimensions beyond those defined by Costa and Silva.

AlNoamany et al. [15, 13, 14] 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. We focus both on being able to re-access these preserved pages but also to aggregate and make accessible personal and private captures (Chapter 7), which was out of the scope of the studies performed by AlNoamany et al.

3.4.2 PRIVATE WEB ARCHIVING

Brunelle et al. [49] 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 dissertation 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. [167] 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. We explore this from a technical standpoint in Chapter 6 to address question RQ4. Rauber et al.'s 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 introduction of a systematic means of access regulation to this private content is addressed in Section 7.1.2.

The Snowden Archive-in-a-Box project [133] 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. In lieu of persistent regulated access to private captures (as facilitated by the dynamics in Section 6.1), replication of this archived Web data is the crux of the project. Our studies and tools for peer-to-peer collaboration and propagation (Section 4.5) would facilitate the goals of this project beyond the scope initially described, further emphasizing the potential for piecemeal adoption of the framework described in Chapter 7.

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 [95]. A site author may provide exclusions to archival content for their domain using robots.txt.

Marshall and Shipman [138] 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 [137] 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. In Section 4.5 we extensively expand on this concept but allow the by-reference "key" in the form of CDXJ indexes to be used as the source of replay and collaboration. This approach facilitates permanence of personal Web archives.

Thelwall and Vaughan [192] 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. [83] 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. Access models beyond the scope of these two works is described in relation to aggregating private and public Web archives in the Access Patterns in Section 7.3.

Marshall [135, 136] 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. As above, we enumerated these access and usage patterns in Section 7.3.

In our previous work [123, 117, 118, 52] 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 [50]. We have preliminarily used browser-based tools

[125] for a subset of the Web archives we created from the private live Web to generate private Web archives. We describe these tools more in-depth in Chapter 4.

Strodl et al. [190] 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 [120] to encourage users to archive their private Web content by removing technical barriers in the preservation software. Strodl et al.'s work abstracts the access issues that is addressed when the implementation of the framework creates data that is 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 dissertation. 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, which we extend on in an accessible way in this dissertation. 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

TOOLS TO ENABLE THE PERSONAL WEB ARCHIVIST

They won't listen. Do you know why? Because they have certain fixed notions about the past. Any change would be blasphemy in their eyes, even if it were the truth. They don't want the truth; they want their traditions.

- Isaac Asimov, *Pebble in the Sky [21]*

As the Web has evolved and society has deemed it culturally significant and thus worth saving, 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 [48]. Facebook.com, for example, only shows temporal details on-demand, obscuring the potentially vast amount of content until it is explicitly requested (Figure 2 in Chapter 1). 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 (Chapter 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. Despite the personal natures of their captures of a site like `cnn.com`, the users may also be willing to allow aggregation of these particular captures (green box).

The issues of the Web being dynamic beyond the capability of institutional tools and content being sensitive and in need of an individual's 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.

This chapter addresses Research Questions 2 and 3 as facilitated by tools built in support of this dissertation. In Section 4.1 we discuss our initial efforts in preserving social

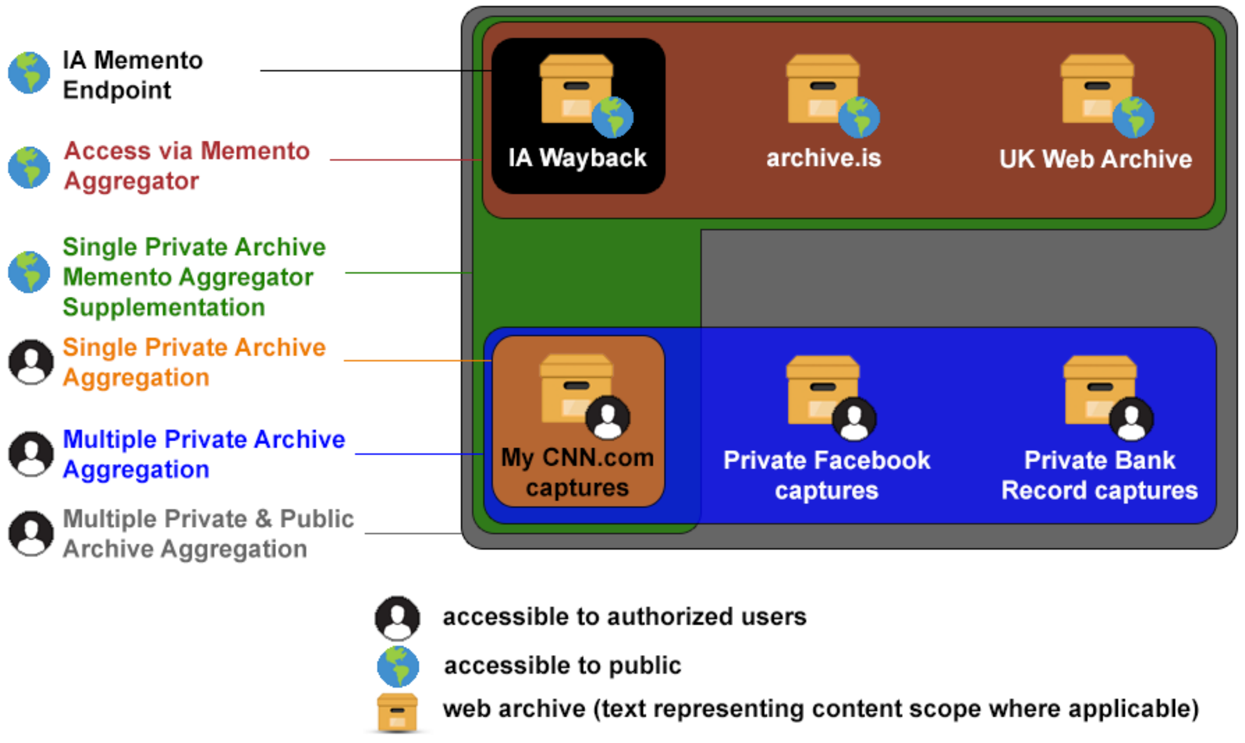

IA Memento Endpoint
Access via Memento Aggregator
Single Private Archive Memento Aggregator Supplementation
Single Private Archive Aggregation
Multiple Private Archive Aggregation
Multiple Private & Public Archive Aggregation
IA Wayback
archive.is
UK Web Archive
My CNN.com captures
Private Facebook captures
Private Bank Record captures
accessible to authorized users
accessible to public
web archive (text representing content scope where applicable)

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.

media content, specially Facebook. Section 4.2 describes our efforts at facilitating a more standards-based approach (creating WARC [98] files) at preserving both social media as well as deep Web content that was otherwise unpreserved. In Section 4.3 we describe a tool to make archive crawling and replay easier for personal Web archivists as well as a second iteration of the tool for higher fidelity archiving than institutional grade crawlers. In Section 4.4 we describe our work on making the live and archived Web more seamless as a more usable mechanism for interaction with the Webs using Memento. In Section 4.5 we describe our software-based efforts for facilitating collaboration of personal and private Web archives using IPFS [29].

4.1 ARCHIVE FACEBOOK

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 [126] 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 [142]), so we created an extensible framework for preserving content behind authentication [108], 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.

4.2 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 [125, 127], 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) [98], 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 developing WARCreate, 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) [204]. 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.

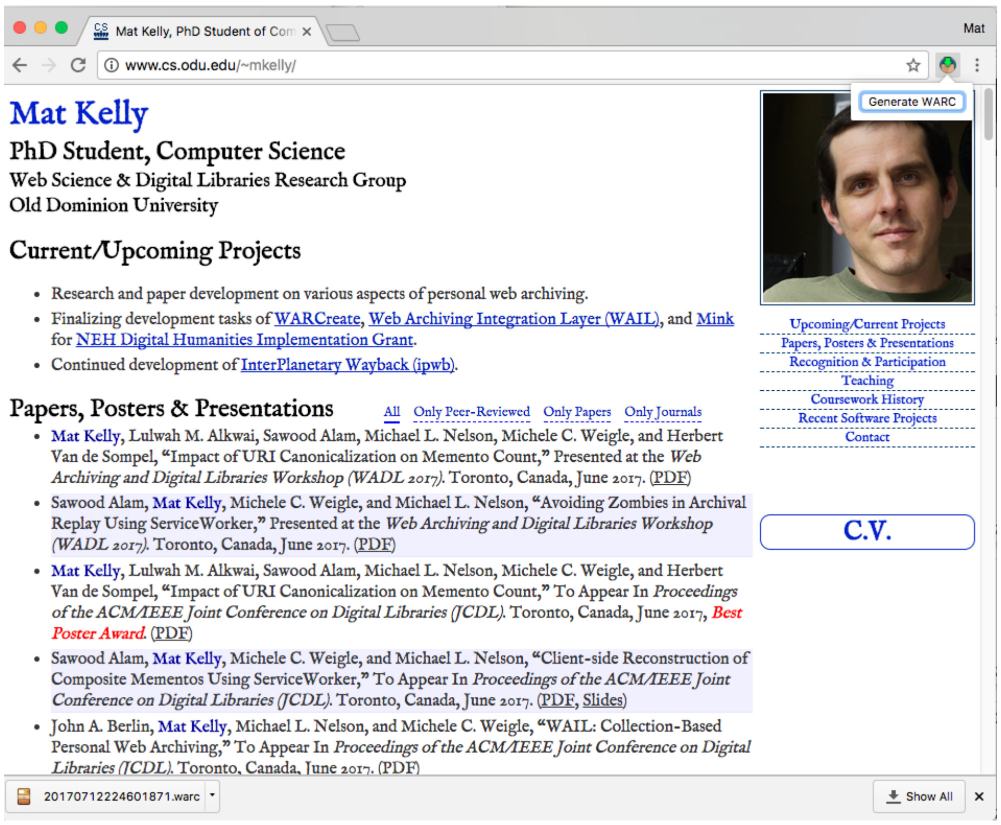
Archiving content from the browser provided a unique perspective to the Web archiving process. Content that is otherwise inaccessible with by-reference delegation (i.e., instructing another tool to archive what is at a URI) could now be preserved. On the other hand, those pages may contain sensitive,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

4.3 WEB ARCHIVING INTEGRATION LAYER (WAIL)

While creating a browser-based tool for capture (Section 4.2), we spun off the server component into a tool (Section 4.3.1) that initially catered to the shortcomings of browsers with the eventual evolution of including capture and replay tools. We further evolved this tool (Section 4.3.2) to use higher fidelity capture and replay methods. In this section we describe the evolution of WAIL.

4.3.1 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 [112] (RQ2). The File API [164], still in the draft stage at the time and not yet



Mat Kelly
PhD Student, Computer Science
Web Science & Digital Libraries Research Group
Old Dominion University

Current/Upcoming Projects

- Research and paper development on various aspects of personal web archiving.
- Finalizing development tasks of WARCreate, Web Archiving Integration Layer (WAIL), and Mink for NEH Digital Humanities Implementation Grant.
- Continued development of InterPlanetary Wayback (ipwb).

Papers, Posters & Presentations All Only Peer-Reviewed Only Papers Only Journals

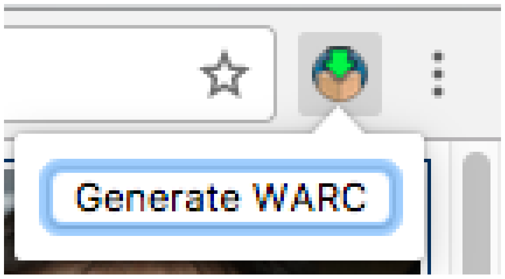
- Mat Kelly, Lulwah M. Alkwai, Sawood Alam, Michael L. Nelson, Michele C. Weigle, and Herbert Van de Sompel, "Impact of URI Canonicalization on Memento Count," Presented at the *Web Archiving and Digital Libraries Workshop (WADL 2017).* Toronto, Canada, June 2017. (PDF)
- Sawood Alam, Mat Kelly, Michele C. Weigle, and Michael L. Nelson, "Avoiding Zombies in Archival Replay Using ServiceWorker," Presented at the *Web Archiving and Digital Libraries Workshop (WADL 2017).* Toronto, Canada, June 2017. (PDF)
- 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," To Appear In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL).* Toronto, Canada, June 2017, *Best Poster Award.* (PDF)
- Sawood Alam, Mat Kelly, Michele C. Weigle, and Michael L. Nelson, "Client-side Reconstruction of Composite Mementos Using ServiceWorker," To Appear In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL).* Toronto, Canada, June 2017. (PDF, Slides)
- John A. Berlin, Mat Kelly, Michael L. Nelson, and Michele C. Weigle, "WAIL: Collection-Based Personal Web Archiving," To Appear In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL).* Toronto, Canada, June 2017. (PDF)

Upcoming/Current Projects
Papers, Posters & Presentations
Recognition & Participation
Teaching
Coursework History
Recent Software Projects
Contact

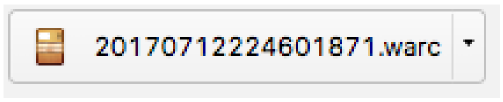
C.V.

20170712224601871.warc Show All

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

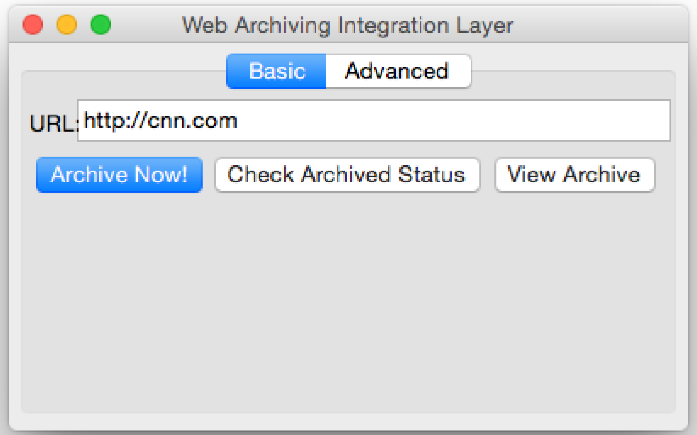


(b) The WARCreate popup consists of a single button interface for simplicity to encourage preservation without complication.

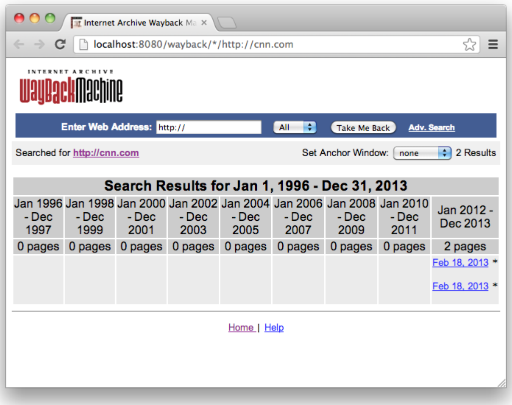


(c) Native Chrome UI provides direct access WARCreate-generated WARC file.

Fig. 31 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).



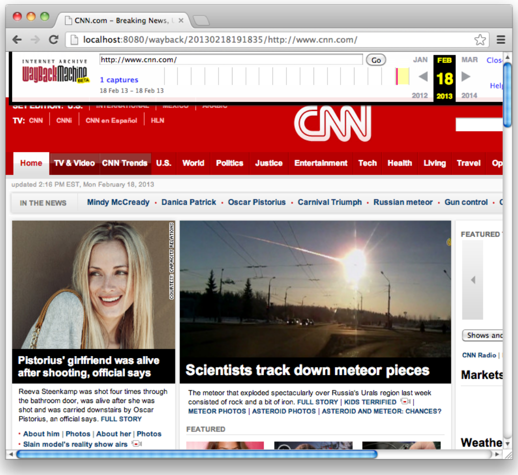
(a) The original WAIL Interface (ca. 2012)


Internet Archive Wayback Ma
localhost:8080/wayback/*/http://cnn.com
INTERNET ARCHIVE WaybackMachine
Enter Web Address: http://
All
Take Me Back
Adv. Search

Searched for http://cnn.com Set Anchor Window: none 2 Results

Search Results for Jan 1, 1996 - Dec 31, 2013								
Jan 1996 - Dec 1997	Jan 1998 - Dec 1999	Jan 2000 - Dec 2001	Jan 2002 - Dec 2003	Jan 2004 - Dec 2005	Jan 2006 - Dec 2007	Jan 2008 - Dec 2009	Jan 2010 - Dec 2011	Jan 2012 - Dec 2013
0 pages	0 pages	0 pages	0 pages	0 pages	0 pages	0 pages	0 pages	2 pages
								Feb 18, 2013 * Feb 18, 2013 *

Home | Help


CNN.com - Breaking News,
localhost:8080/wayback/20130218191835/http://www.cnn.com/
INTERNET ARCHIVE WaybackMachine
http://www.cnn.com/
Go
1 captures
18 Feb 13 - 18 Feb 13
JAN FEB MAR
18
2012 2013 2014
CNN
Home TV & Video CNN Trends U.S. World Politics Justice Entertainment Tech Health Living Travel
IN THE NEWS Mindy McCready · Danica Patrick · Oscar Pistorius · Carnival Triumph · Russian meteor · Gun control
Pistorius' girlfriend was alive after shooting, official says
Scientists track down meteor pieces
FEATURED
Markets
Weather

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

(c) Viewing CNN capture in local OpenWayback

Fig. 32 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.

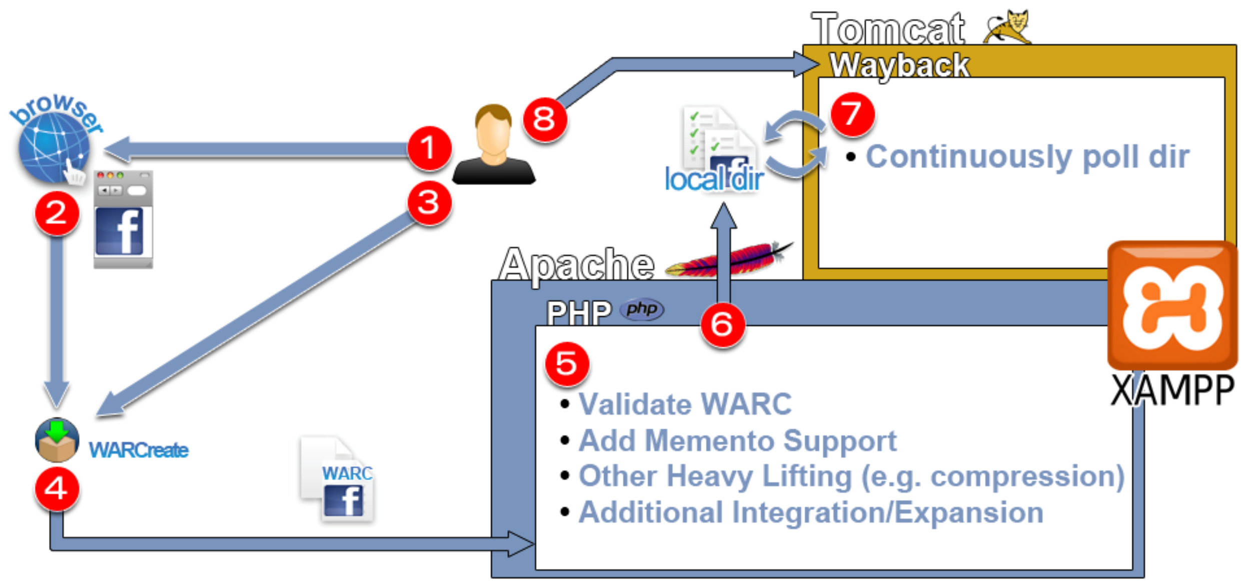


Fig. 33 When developing WARCreate [125], a local server instance was originally required to write to the file system. When browsers became more capable, the server components were repackaged along with the additional inclusion of Heritrix and deployed as Web Archiving Integration Layer (WAIL) [120, 121].

fully supported by any browser [62], also provided no reprieve, as files produced by browser extensions were sandboxed and inaccessible from the rest of the file system.

To counter browser shortcomings at the time, rather than rely on a central Web-based endpoint, we leveraged the cross-platform and desktop-based XAMPP [67] to act as a “local server” bridge to allow the user to write WARCs to their local file system (Figure 33) by performing an HTTP POST [79] with the WARC contents to the “server”. In doing so, we also enabled users to use desktop-based applications relating to Web archiving. For example, to enable users to replay the WARCs they created with WARCreate (Section 4.2) or from other sources, we configured and adapted OpenWayback to utilize the Tomcat [199] runtime included in XAMPP. We removed superfluous portions of XAMPP (e.g., service control interface code) and eventually extracted the Tomcat setup to use OpenWayback as the sole application.

Once browsers became more capable of interacting with the local file system outside of a limited “sandbox”, the server component was unnecessary for WARCreate to generate WARCs that could be saved to the local system beyond the sandbox. As we had leveraged

OpenWayback for WARC replay, we also bundled Heritrix for conventional, institutional-grade Web crawling (cf. WARCreate's page-at-a-time archiving) and rewrote a tailored Graphical User Interface (GUI) to create Web Archiving Integration Layer (WAIL) [120, 121]. The interface was originally programmed in Python 2.7 using wxPython [166], a Python port of wxWidgets [187], a cross-language and cross-platform UI library. The Python code was compiled to a native executable (`.app` on macOS/MacOS X[1] and `.exe` on Windows) using PyInstaller [206], a program that performs the latter task from Python scripts. The simple, graphical, and native interface in WAIL encouraged users to create personal Web archives without the required technical overhead and with the ease of entering a URI and selecting an "Archive Now!" button (Figure 32).

4.3.2 WAIL-ELECTRON

While the original WAIL bundled Heritrix, OpenWayback, and other institution-grade Web archiving tools, these tools were often incapable of capturing certain Web content, e.g., content using certain JavaScript features (see Chapter 5). The original WAIL (Section 4.3.1) was built by compiling a set of Python scripts to a native application, relative to the set of supported platforms (MacOS X and Windows). Accounting for the nuances in the bundled tools as well as writing cross-platform GUI-driven code from a Python script proved to be difficult to maintain. Further, relying on the capability of the bundled tools (and thus sometimes the incompleteness of captures of URI-Rs) was a fundamental shortcoming of captures generated by personal Web archivists. One additional aspect beyond the bundled tools that limited WAIL was the inability to organize captures into collections. The original WAIL assumed a single collection where every capture was accessible at a single Web-based endpoint.

We reimagined WAIL [32] as an Electron[2] application. Electron is a framework that allows conventional Web technologies like HTML, CSS, and JavaScript to be compiled to a variety of native applications. Applications leveraging Electron may display a GUI with much of the same markup and style that would produce a rendered Web page when viewed in a Web browser. In lieu of OpenWayback, we opted to integrate pywb [129] for archival replay. pywb provides native collection management as well as account for some advanced archival replay dynamics not present in OpenWayback. As with the original WAIL, we provided

[1]During the research for this dissertation, Apple Inc. changed the name of their operating system from "MacOS X" to "OS X" to "macOS". While these are semantically interchangeable, the variant in this dissertation is representative of the name at the relative time.

[2]`https://electronjs.org/`

Heritrix as an option for preservation but unlike the original, we additionally included integration with a native Chromium browser into the archiving process. This additional integration allowed previously missed Web content to be surfaced for preservation from sites like Twitter to produce a more accurate and comprehensive Web archive [32]. WAIL-Electron's collection-based archiving first asks users to specify a set of seeds to initially crawl for the collection, similar to the commercial Archive-It service provided by Internet Archive. WAIL also allows a user to supply descriptions and metadata for the collection to give the captures semantic scope.

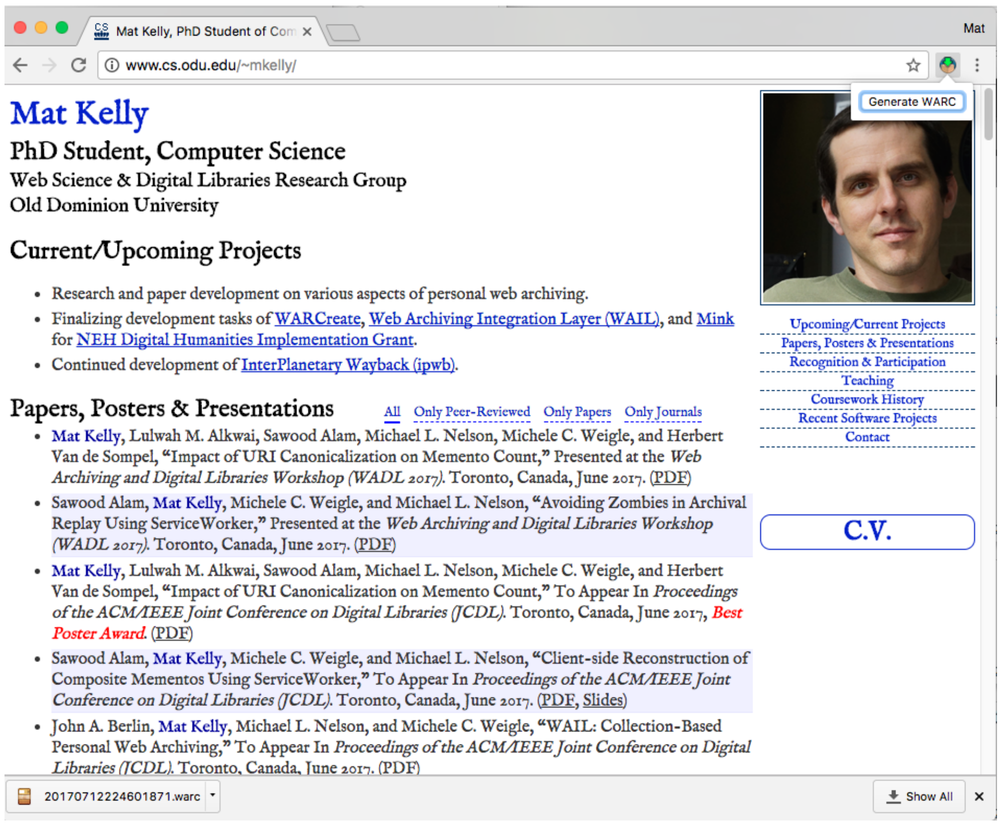
Collections

Subscription-based Web archiving services like Archive-It (Appendix A) allow users with limited technical knowledge to create and replay personalized collections of Web archives. Archive-It provides its users with a simple interface to create collections and to launch complex archival crawls. Similarly, Webrecorder[3] allows any user to register for the service and provides them with the ability to create and manage personalized collections of Web archives. But unlike Archive-It, Webrecorder requires its user to manually drive the preservation process or upload content for replay while only providing its users up to five gigabytes of storage. Individuals that wish to freely (*gratis* and *libre*) archive Web pages without arbitrary restrictions beyond the limitations of their personal computers using institutional grade tools must set up an archival Web crawler (e.g., Heritrix) and replay system (e.g., Wayback), time consuming and technical tasks potentially beyond the individual's skill level.

The collection view in WAIL-Electron (Figure 35a) displays an overview of the collections that the software is currently managing and information about them. This information includes the number of seeds contained in the collection along with the collection's size and the last time it was updated.

A user may easily create a new collection by clicking the "New Collection" button. Doing so displays a dialog (Figure 35b), prompting the user for a collection name, title, and description. These values are propagated to the WAIL interface and are viewable when replaying the collection through Wayback. When viewing a collection, WAIL displays metadata about each seed in the collection (e.g., date added, last time archived, number of captures) and a direct means to replay the seed using the Wayback instance provided. Users may also add seeds to be crawled through the WAIL interface or may import any previously existing

[3]`https://webrecorder.io/`



Mat Kelly
PhD Student, Computer Science
Web Science & Digital Libraries Research Group
Old Dominion University

Current/Upcoming Projects

- Research and paper development on various aspects of personal web archiving.
- Finalizing development tasks of WARCreate, Web Archiving Integration Layer (WAIL), and Mink for NEH Digital Humanities Implementation Grant.
- Continued development of InterPlanetary Wayback (ipwb).

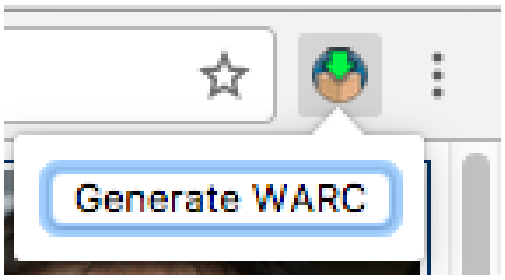
Papers, Posters & Presentations   All Only Peer-Reviewed Only Papers Only Journals

- 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," Presented at the *Web Archiving and Digital Libraries Workshop (WADL 2017)*. Toronto, Canada, June 2017. (PDF)
- Sawood Alam, Mat Kelly, Michele C. Weigle, and Michael L. Nelson, "Avoiding Zombies in Archival Replay Using ServiceWorker," Presented at the *Web Archiving and Digital Libraries Workshop (WADL 2017)*. Toronto, Canada, June 2017. (PDF)
- 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," To Appear In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. Toronto, Canada, June 2017, *Best Poster Award*. (PDF)
- Sawood Alam, Mat Kelly, Michele C. Weigle, and Michael L. Nelson, "Client-side Reconstruction of Composite Mementos Using ServiceWorker," To Appear In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. Toronto, Canada, June 2017. (PDF, Slides)
- John A. Berlin, Mat Kelly, Michael L. Nelson, and Michele C. Weigle, "WAIL: Collection-Based Personal Web Archiving," To Appear In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. Toronto, Canada, June 2017. (PDF)

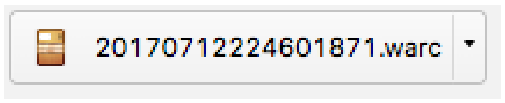
Upcoming/Current Projects
Papers, Posters & Presentations
Recognition & Participation
Teaching
Coursework History
Recent Software Projects
Contact

C.V.

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

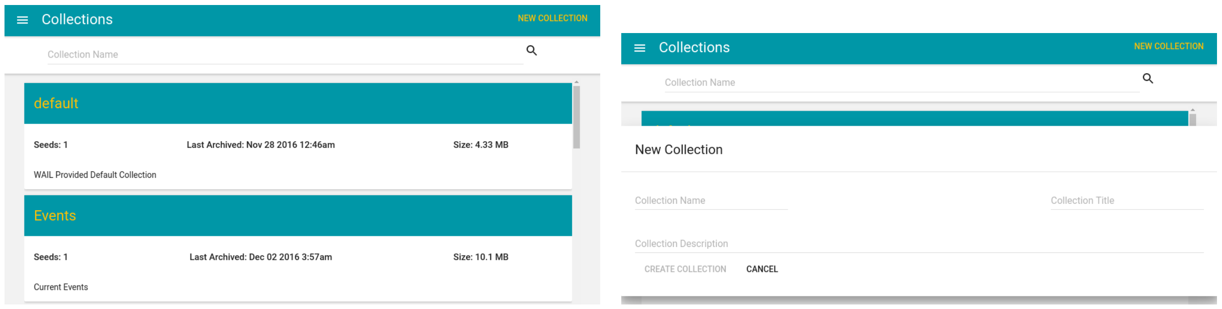


(b) The WARCreate popup consists of a single button interface for simplicity to encourage preservation without complication.



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

Fig. 34 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).



default

Seeds: 1 Last Archived: Nov 28 2016 12:46am Size: 4.33 MB

WAIL Provided Default Collection

Events

Seeds: 1 Last Archived: Dec 02 2016 3:57am Size: 10.1 MB

Current Events

(a) Collection View **(b)** Collection Creation

Fig. 35 WAIL-Electron provides a set of interfaces for creating and managing collections of Web archives within the application.

WARC [98] files that were generated from any source. WAIL also provides a mechanism to check on the state of the resource on the live Web from within the same interface prior to the user initiating a crawl.

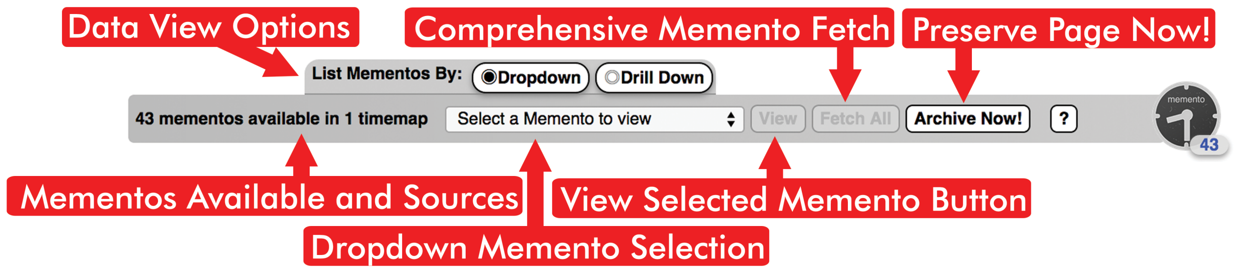
After entering the URI for a new seed, WAIL automatically configures and launches the crawl. While a crawl or a set of crawls is underway, WAIL provides real-time progress monitoring for the crawls within the interface. Upon completion of a crawl, WAIL will automatically associate the generated WARC file to the collection and ensure its ingestion by Wayback. As an additional feature, WAIL provides an interface for a user to monitor and archive Twitter content automatically. Users may specify criteria by which to identify tweets to be archived and to which collection the preserved tweet is added. Once the monitoring has started and a tweet has been identified for archiving, the tweet will be archived and automatically added to the specified collection.

4.4 MINK

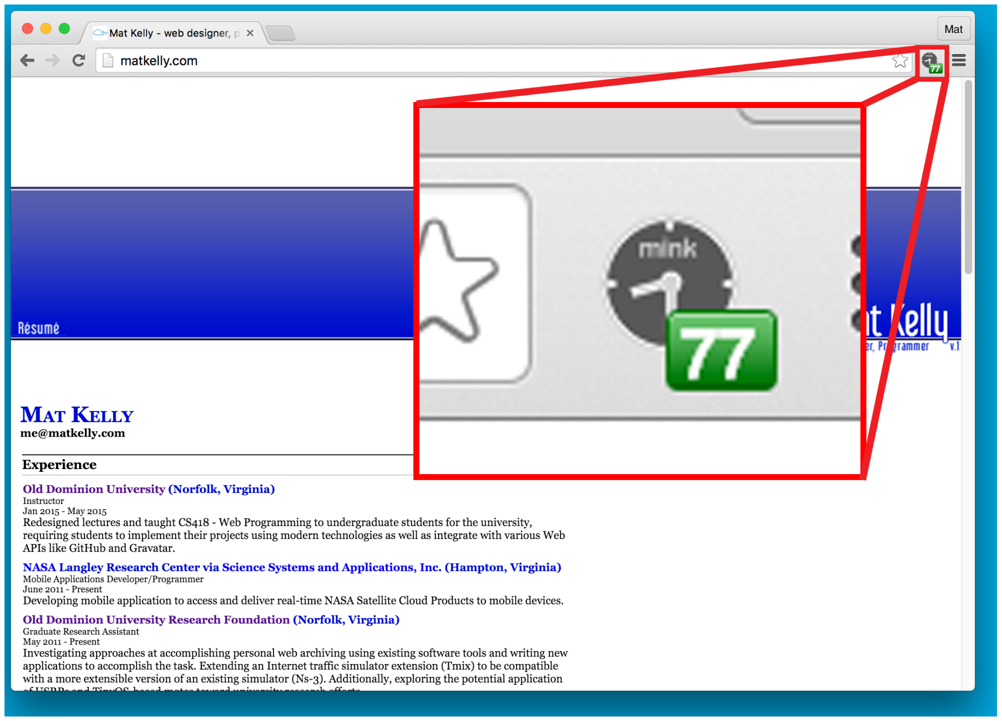
Building upon our work in enabling Web users to preserve content, we inverted our perspective on tool building to the realm of access of Web archives. The temporal gaps in only using a single Web archive as a source for the historical record (e.g., looking solely to Internet Archive for the Web's history) may be mitigated by including additional Web archives. 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 [122] (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 36a) but refined to be less obtrusive in the browser button bar (Figure 36b)) 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 [116, 115]). 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 36c and 36c, 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 [106], 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.

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 (Section 4.1), 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 [10], 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 dissertation.

To relate back to *personal* Web archiving and to make Mink more useful for individual archivists, we later expanded on Mink [204] 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 captures aggregated inline with institutions' 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.


Data View Options
Comprehensive Memento Fetch
Preserve Page Now!
List Mementos By:
Dropdown
Drill Down
43 mementos available in 1 timemap
Select a Memento to view
View
Fetch All
Archive Now!
?
memento
43
Mementos Available and Sources
View Selected Memento Button
Dropdown Memento Selection

(a)



Mat Kelly
me@matkelly.com

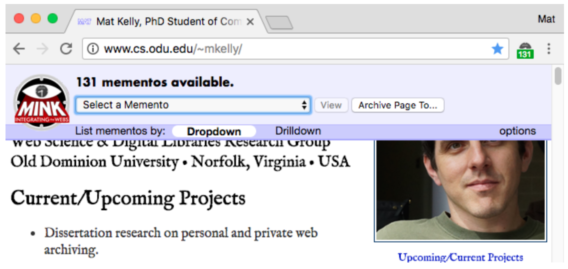
Experience

Old Dominion University (Norfolk, Virginia)
Instructor
Jan 2015 - May 2015
Redesigned lectures and taught CS418 - Web Programming to undergraduate students for the university, requiring students to implement their projects using modern technologies as well as integrate with various Web APIs like GitHub and Gravatar.

NASA Langley Research Center via Science Systems and Applications, Inc. (Hampton, Virginia)
Mobile Applications Developer/Programmer
June 2011 - Present
Developing mobile application to access and deliver real-time NASA Satellite Cloud Products to mobile devices.

Old Dominion University Research Foundation (Norfolk, Virginia)
Graduate Research Assistant
May 2011 - Present
Investigating approaches at accomplishing personal web archiving using existing software tools and writing new applications to accomplish the task. Extending an Internet traffic simulator extension (Tmix) to be compatible with a more extensible version of an existing simulator (Ns-3). Additionally, exploring the potential application

(b)

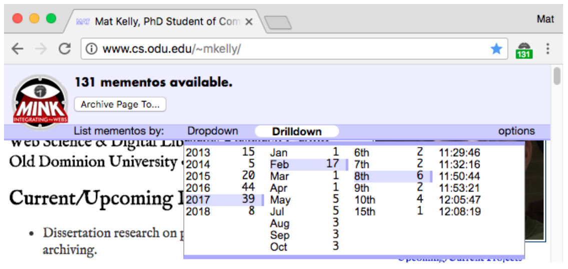


Web Science & Digital Libraries Research Group
Old Dominion University • Norfolk, Virginia • USA

Current/Upcoming Projects

- Dissertation research on personal and private web archiving.

(c)



2013	15	Jan	1	6th	2	11:29:46
2014	5	Feb	17	7th	2	11:32:16
2015	20	Mar	1	8th	6	11:50:44
2016	44	Apr	1	9th	2	11:53:21
2017	39	May	5	10th	4	12:05:47
2018	8	Jul	5	15th	1	12:08:19
		Aug	3			
		Sep	3			
		Oct	3			

(d)

Fig. 36 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).

Mink was then able to select a source of aggregation but did not exhibit sufficient control as to the sources of aggregation, i.e., what archives the requested aggregator aggregated. In Section 6.1 we discuss and provide a more systematic solution for client-side archival specification. In Section 8.3.4 we expand on the original concept and implementation of Mink to cater more to the additional functionality and roles of the framework described in Chapter 7.

4.5 PEER-TO-PEER COLLABORATION AND PROPAGATION

Unlike institutional Web archives, personal Web archives often simply reside on a user's machine. This is problematic when a machine fails, among other circumstances, so creating copies of the captures helps to facilitate its accessibility in the future. We developed InterPlanetary Wayback (ipwb) [113, 7] to propagate Web archive content into the peer-to-peer InterPlanetary File System (IPFS) [29] 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 Section 7.1.2.

ipwb first performs an "indexing" procedure on a target set of WARCs, as specified by the user. This indexing procedure initially entails iterating through and extracting the contents of `warc-response` records (e.g., Figures 18d and 18e) in the set of WARC files. The contents of the records are added to IPFS via ipwb (Figure 37), generating unique content-derived hash (Content-Identifier or CID) that allows the contents of the record to be retrieved in IPFS by this CID. For example, Figure 38 shows two identical image resource representations that can be accessed on the live Web at different URIs. Adding either of these files to IPFS will produce identical accessible CIDs despite the image residing at different URIs. The resource representation can be fetched using this CID, which itself is representative of the content, independent of the URI where it originally resided. Changing this image and re-adding to IPFS would produce a different CID. Retaining the association of URI-R to a resource is important to retain the original context in Web archiving. This loss of association inherent in IPFS is mitigated in ipwb through an associative indexing procedure, described below.

ipwb retains the CID "locators" produced from adding the contents of the WARC records and associates them with a URI-R and datetime (as extracted from the WARC source) combination via CDXJ (Section 2.4.4) for retrieval (example ipwb CDXJ shown in Figure 41). For a user to propagate the content of their archive, they first add the relevant contents

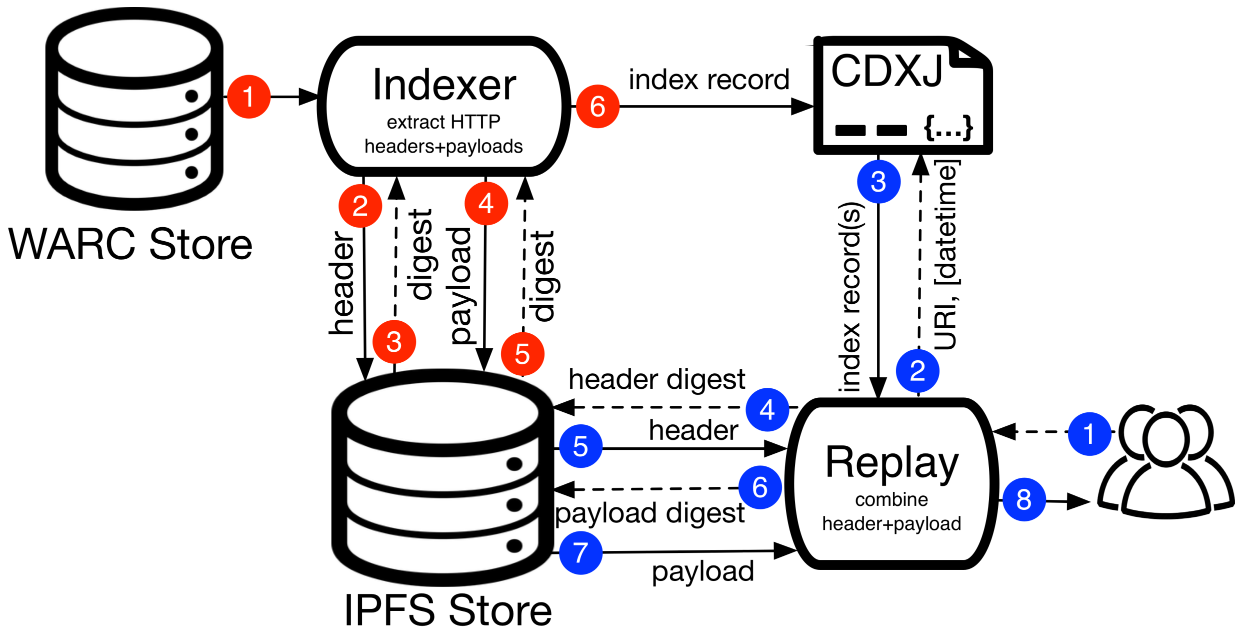
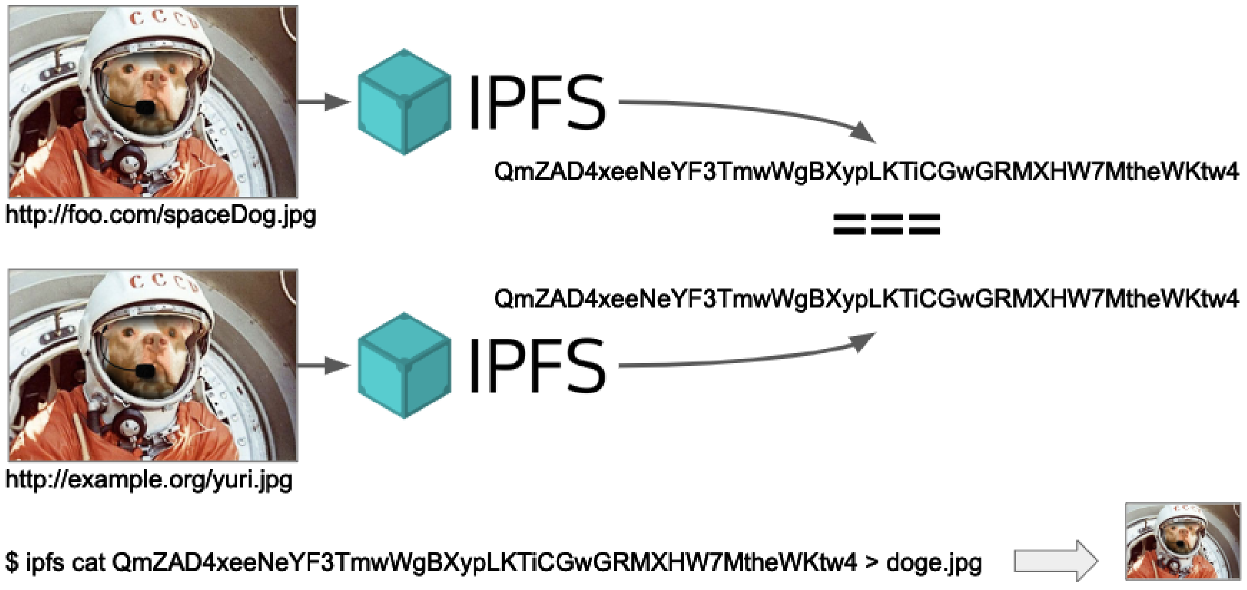


Fig. 37 Pushing WARC records to IPFS (red circles) requires the WARC response headers and payloads to be extracted (red 1), pushed to IPFS to obtain digest hashes (red 2-5), and hashes to be included in an index (red 6). The replay process (blue circles) has a user querying a replay system as usual (blue 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).


IPFS
QmZAD4xeeNeYF3TmwWgBXypLKTiCGwGRMXHW7MtheWKtw4
http://foo.com/spaceDog.jpg
===
QmZAD4xeeNeYF3TmwWgBXypLKTiCGwGRMXHW7MtheWKtw4
IPFS
http://example.org/yuri.jpg

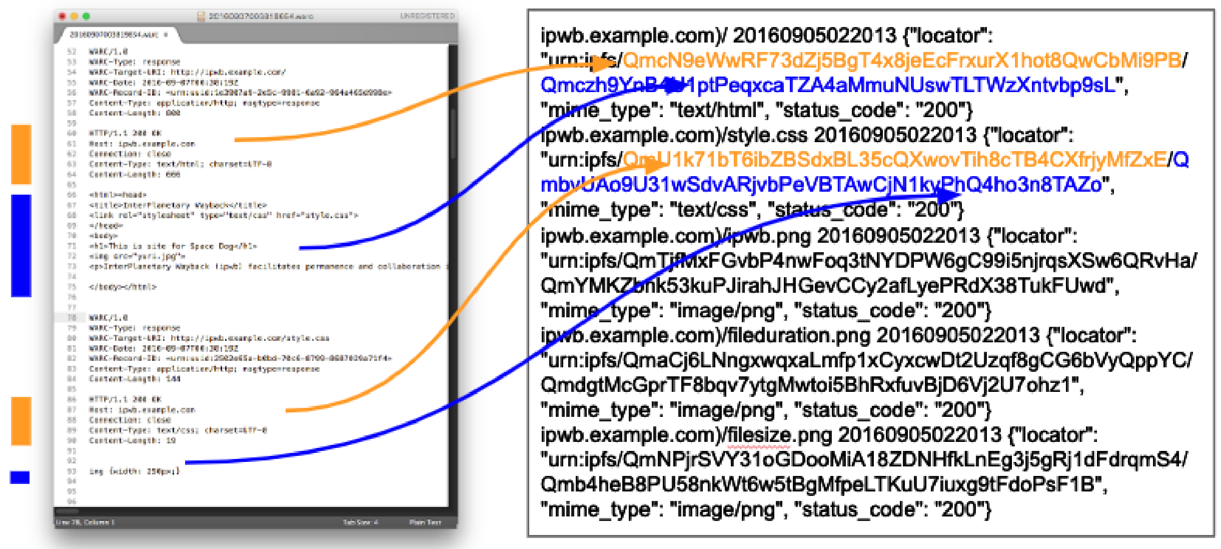
```
$ ipfs cat QmZAD4xeeNeYF3TmwWgBXypLKTiCGwGRMXHW7MtheWKtw4 > doge.jpg
```

Fig. 38 Content addressing entails generating a hash of a file and using that hash as a means up retrieving the content. Two identical images at different URIs will result in the same hash when content addressing and identical hashing algorithms are are used. In IPFS, this hash can be used for retrieving the image instead of retrieving the image by URI.

to IPFS using ipwb, thus generating a CDXJ index with the respective set of associated URI-Rs, datetimes, and CIDs (Figure 39). The user then may share this CDXJ file, which can be used as the basis for lookup and retrieval of captures from the IPFS network. Another user would direct their ipwb instance to use the received CDXJ file, access the replay interface, and navigate to a URI-M, whose contents are fetched from IPFS using the ipwb component. This process can be performed en-masse to propagate Web archives to facilitate distributed preservation through redundancy.

ipwb provides a Web archive replay interface much in the same way as OpenWayback and pywb are a replay interface to a set of WARCs. On accessing the interface (Figure 40) using a Web browser, the set of URI-Rs in the archive are displayed in a datestamped list. A text input box also provides a means for a user to lookup the set of captures in their ipwb instance, as executed using a CDXJ index, by URI-R and provides a conventional navigation experience to the captures.

As we progressively built ipwb over time, we explored more modern approaches as rewriting and navigation. In most Web archive replay systems, the archived contents of a Web



```
WARC/1.0
WARC-Type: response
WARC-Target-URI: http://ipwb.example.com/
WARC-Date: 2016-09-07T00:38:19Z
WARC-Record-ID: <urn:uuid:1e2087a8-2e5c-9981-4a92-984a465d098e>
Content-Type: application/http; msgtype=response
Content-Length: 808

HTTP/1.1 200 OK
Host: ipwb.example.com
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 666

<html><head>
<title>InterPlanetary Wayback</title>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
<h1>This is site for Space Dog</h1>
<img src="yuri.jpg">
<p>InterPlanetary Wayback (ipwb) facilitates permanence and collaboration i

</body></html>


WARC/1.0
WARC-Type: response
WARC-Target-URI: http://ipwb.example.com/style.css
WARC-Date: 2016-09-07T00:38:19Z
WARC-Record-ID: <urn:uuid:2502e65a-b6bd-78c6-8799-0687026a71f4>
Content-Type: application/http; msgtype=response
Content-Length: 144

HTTP/1.1 200 OK
Host: ipwb.example.com
Connection: close
Content-Type: text/css; charset=UTF-8
Content-Length: 19


img {width: 250px;}
```

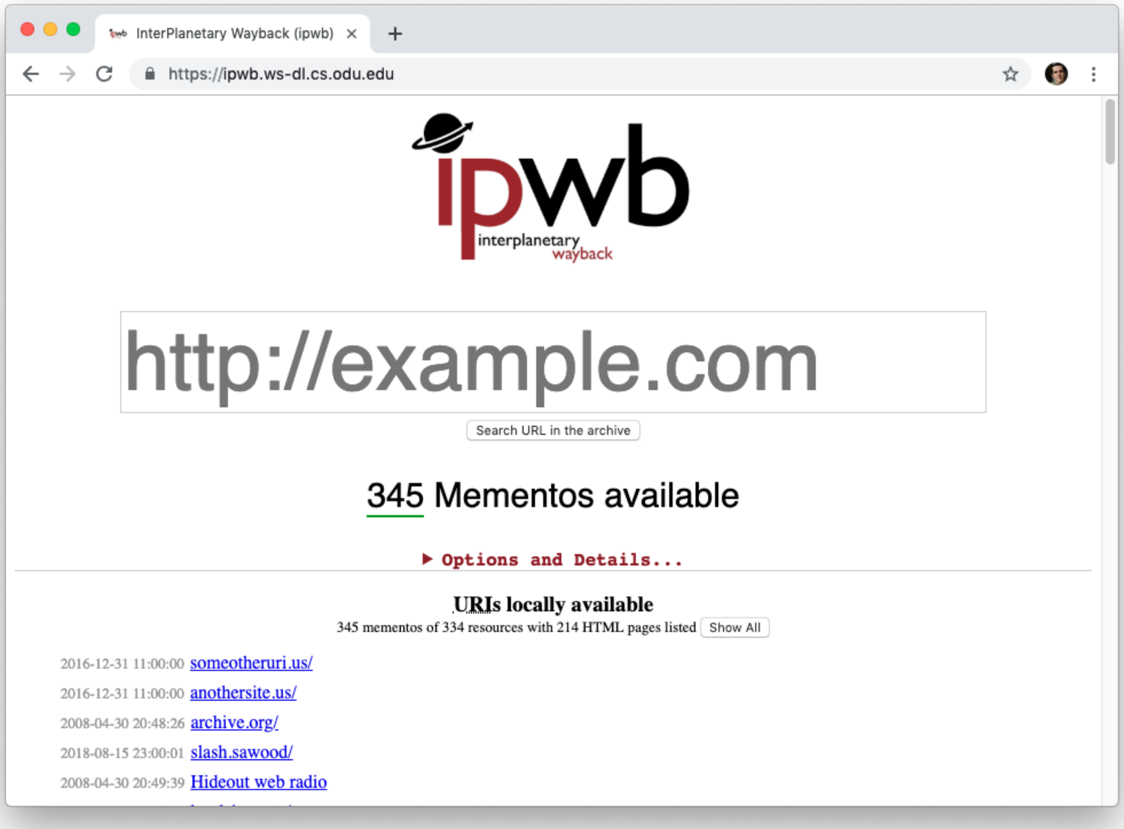
```
ipwb.example.com)/ 20160905022013 {"locator":
"urn:ipfs/QmcN9eWwRF73dZj5BgT4x8jeEcFrxurX1hot8QwCbMi9PB/
Qmczh9YoB4M1ptPeqxcaTZA4aMmuNUswTLTWzXntvbp9sL",
"mime_type": "text/html", "status_code": "200"}
ipwb.example.com)/style.css 20160905022013 {"locator":
"urn:ipfs/QmU1k71bT6ibZBSdxBL35cQXwovTih8cTB4CXfrjyMfZxE/Q
mbvUAo9U31wSdvARjvbPeVBTAwCjN1kyPhQ4ho3n8TAZo",
"mime_type": "text/css", "status_code": "200"}
ipwb.example.com)/ipwb.png 20160905022013 {"locator":
"urn:ipfs/QmTjfMxFGvbP4nwFoq3tNYDPW6gC99i5njrqsXSw6QRvHa/
QmYMKZbnk53kuPJirahJHGevCCy2afLyePRdX38TukFUwd",
"mime_type": "image/png", "status_code": "200"}
ipwb.example.com)/fileduration.png 20160905022013 {"locator":
"urn:ipfs/QmaCj6LNngxwqxaLmfp1xCyxcwDt2Uzqf8gCG6bVyQppYC/
QmdgtMcGprTF8bqv7ytgMwtoi5BhRxfuvBjD6Vj2U7ohz1",
"mime_type": "image/png", "status_code": "200"}
ipwb.example.com)/filesize.png 20160905022013 {"locator":
"urn:ipfs/QmNPjrSVY31oGDooMiA18ZDNHfkLnEg3j5gRj1dFdrqmS4/
Qmb4heB8PU58nkWt6w5tBgMfpeLTKuU7iuxg9tFdoPsF1B",
"mime_type": "image/png", "status_code": "200"}
```

Fig. 39 ipwb generates hashes by extracting HTTP headers and HTTP entity bodies from WARC files (denoted with orange and blue bars on left, respectively) and generates a separate hash for each. These two hashes are prefixed with a `urn:ipfs/` locator, added as a value for the `locator` JSON attribute, and associated with a SURTed URI-R and datetime corresponding to the warc-response record from which it was extracted.

page would normally contain links to the live Web, as represented in the memento. Conventional replay systems "rewrite" these links by replacing the contents of the payload with links that point back into the archive to allow for a user to navigate around an archive. ServiceWorkers [179] are a concept modern to the Web[4] that provide a browser-native means of intercepting requests. We leveraged this technology in subsequent versions of ipwb to intercept requests for embedded resources to be "rerouted" to the ipwb instance without affecting the contents of the Web page representation [8]. Whereas conventional Web archive replay systems change the representation to rewrite URIs of embedded resources and links to other captures, ipwb "reroutes" the URIs of the embedded resources and links to provide a more accurate representation by utilizing ServiceWorkers.

Content may be retrieved from IPFS, even without ipwb, using the CID 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 [119], we extended ipwb

[4]With full support from IE Edge, Firefox, Chrome, Safari, and Opera beginning in April 2018 [63]



URIs locally available

345 mementos of 334 resources with 214 HTML pages listed

2016-12-31 11:00:00 someotheruri.us/
2016-12-31 11:00:00 anothersite.us/
2008-04-30 20:48:26 archive.org/
2018-08-15 23:00:01 slash.sawood/
2008-04-30 20:49:39 Hideout web radio

Fig. 40 InterPlanetary Wayback's replay interface provides direct access to URI-Rs over time as well as a search interface.

```
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"
}
```

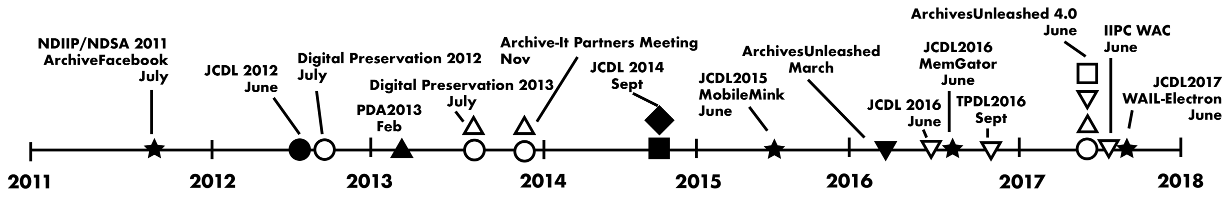
Fig. 41 A CDXJ index allows a memento to be resolved to a WARC record in a playback system. In the ipwb prototype we extract the relevant values from the HTTP response headers at time of index and include the IPFS hashes as the means for a replay system to obtain the HTTP headers and payload corresponding to the URI-M requested.

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 CIDs 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 CID.

4.6 SUMMARY AND CONTRIBUTIONS TO RESEARCH QUESTIONS

In this chapter we described tools we created (Figure 42) to enable individuals to create, manage, interact with, and share personal Web archives. In Section 4.2 we described a browser extension that identified content that was difficult to capture for preservation from a Web browser and enabled users to do so into the WARC format (RQ1). This tool leveraged Web browsers APIs to enable functionality of preservation normally beyond the scope of archival crawlers (RQ2). This tool also partially answered RQ3 with respect to capturing the content behind authentication but left open the question about how these captures can be adequately replayed with respect to the contents they contain.

Section 4.3 introduced bundling institutional grade Web archiving tools like OpenWayback and Heritrix to allow individuals to preserve content from their desktop using a graphical user interface (WAIL). Using the abstraction of a native application instead of having to manually configure these tools provided an easier mechanism for users to crawl the Web sites they felt were worthy of preservation. While the preservation and replay tools are



	WARCreate [125, 127, 121, 204]	WAIL [120, 121, 112, 32]	Mink [122, 106]	ipwb [7, 113, 119]	Archival Acid Test [123]	Others' Tools [10, 106, 32]
Initial Presentation	●	▲	■	▼	◆	★
Supporting Presentation/ Tool Advancement	○	△	□	▽	◇	☆

Fig. 42 In the course of developing this dissertation, we developed and extended numerous tools. Unlike most research software, these tools were publicly released and continually maintained. These tools provided the basis for extension, as applicable, to exhibit the roles of the mementities detailed further in Chapter 7.

institutional-grade, these tools would sometimes create incomplete captures due to shortcomings in their capabilities. We describe the impact that these shortcomings have in our experiments with evaluating archivability in Chapter 5. Following these investigations, we also created an Electron version (Section 4.3.2) of WAIL that allowed our crawls to use a headless browser for preservation – enhancing the capability of the conventional Web archiving process. This progression further remedied the issue of content being difficult to capture (RQ1) by leveraging browser APIs in the crawler process through the Electron abstraction.

Section 4.5 described ipwb, a tool we created and refined to integrate Web archiving with the InterPlanetary File System. Our expansion of this tool's original functionality provided an encryption workflow novel to Web archiving and provided a method, using the extensibility of CDXJ TimeMaps, for the replay system to signal that captures are private and may require special handling (RQ4).

CHAPTER 5

MEASURING ARCHIVABILITY

Impossible. Perhaps the archives are incomplete.

- Obi-Wan Kenobi

If an item does not appear in our records, it does not exist.

- Jocasta Nu, *Star Wars: Episode II Attack of the Clones [65]*

The ease of archiving a Web page (the *archivability*) is impacted by the migration from Web pages to Web applications [52]. Being able to evaluate *archivability* from what was preserved and what is currently preservable (RQ1) with state-of-the-art archiving tools gives a basis for further challenges to be addressed in Web archiving by both institutions and individuals. In this chapter we describe three separate investigations:

- An evaluation of the change in archivability over time [118] (Section 5.1)
- An investigation to evaluate the impact of JavaScript on archivability [52] (Section 5.2)
- An "archival acid test" to determine the state of the art of institutional preservation systems [123] (Section 5.3)

5.1 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 [118] 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 3). 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

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 3 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 [118].

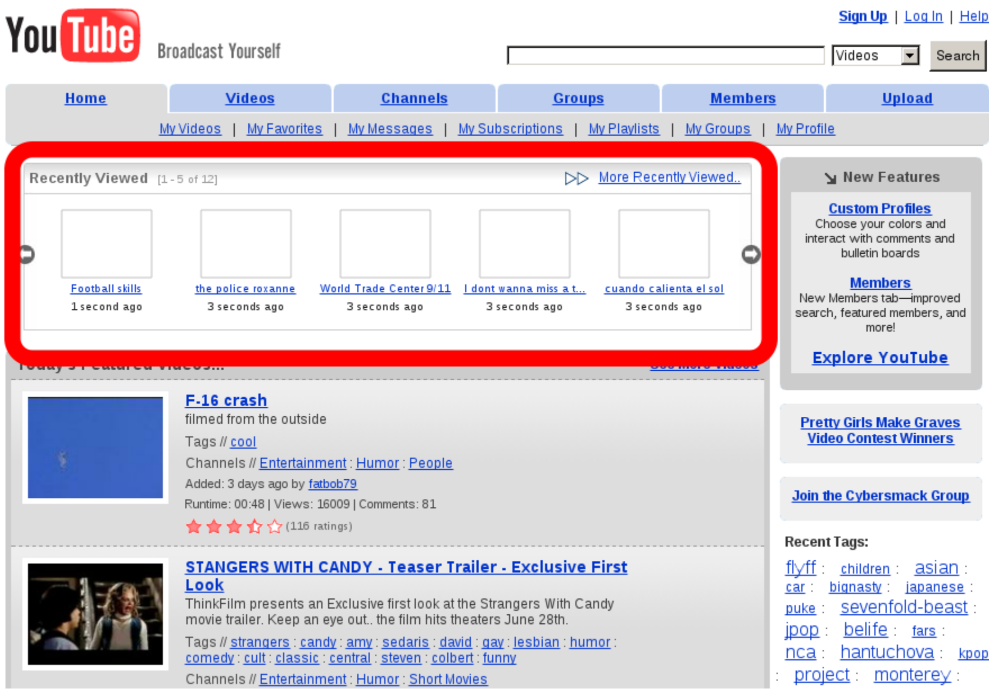
impact on archivability has been profound (see upcoming Section 5.2). Figure 43 shows a capture of `youtube.com` from 2006 with a subtle distinction (circled in red) in the display when JavaScript is enabled (Figure 43a) and disabled (Figure 43b) at the time of capture. The AJAX spinner (above each "loading" message (Figure 43b) 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 Figure 43a) 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.

Figure 44 shows the same URI-R as Figure 43 but from a capture in 2011 (Figure 44a) 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 44b). The browser console at the time of replay (Figure 44b) 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 Figure 44b) 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 Figure 44b, exacerbate the display issue.

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 [196] accessibility compliance for Web sites (and other Web accessibility initiatives [200, 57]) to indicate the trend of the number of missing resources for each URI-R over time (Figure 45). 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 dip in the plot of `nasa.gov` correlates with the annual screenshots in Figure 6 (Chapter 1). The preservation of the White House Web page (Figure 45b) 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. From this we concluded that 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 during 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).

5.2 IMPACT OF JAVASCRIPT ON ARCHIVABILITY

With the recognition that archivability has changed as Web technologies evolved, we continued our investigation with a focus on the impact that JavaScript has on the archivability of Web pages [52]. 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 46 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, allowing the user to plan. The April 2012 version of the page is missing UI elements and functionality (circled)


Sign Up | Log In | Help
Broadcast Yourself
Videos
Search
Home
Videos
Channels
Groups
Members
Upload
My Videos | My Favorites | My Messages | My Subscriptions | My Playlists | My Groups | My Profile

Recently Viewed [1 - 5 of 12] More Recently Viewed..

Football skills 1 second ago
the police roxanne 3 seconds ago
World Trade Center 9/11 3 seconds ago
I dont wanna miss a t... 3 seconds ago
cuando calienta el sol 3 seconds ago

F-16 crash
filmed from the outside
Tags // cool
Channels // Entertainment : Humor : People
Added: 3 days ago by fatbob79
Runtime: 00:48 | Views: 16009 | Comments: 81
(116 ratings)

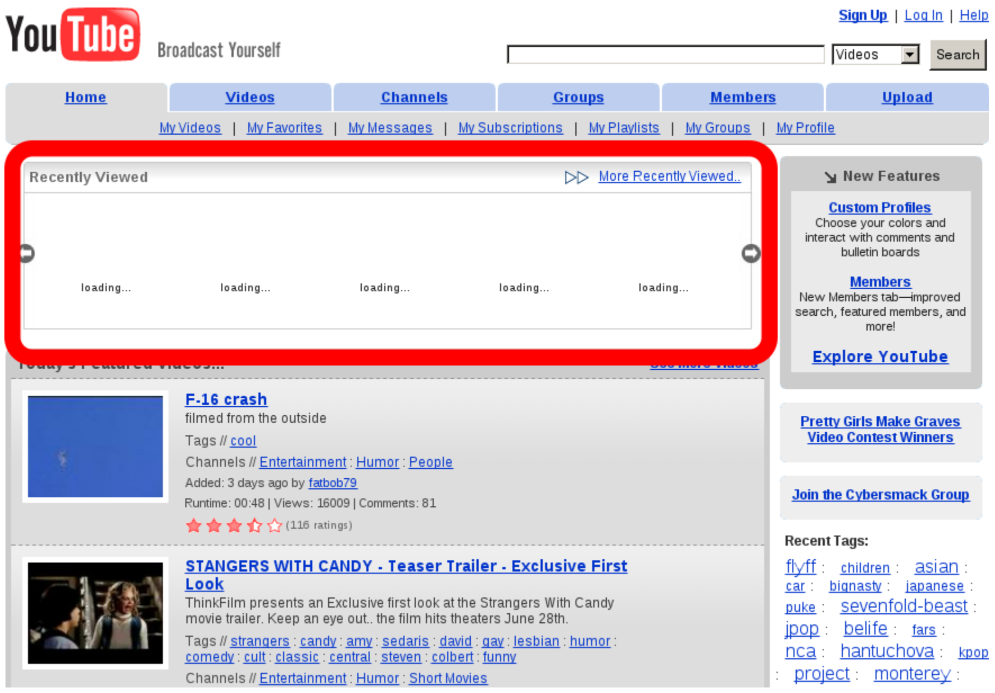
STANGERS WITH CANDY - Teaser Trailer - Exclusive First Look
ThinkFilm presents an Exclusive first look at the Strangers With Candy movie trailer. Keep an eye out.. the film hits theaters June 28th.
Tags // strangers : candy : amy : sedaris : david : gay : lesbian : humor : comedy : cult : classic : central : steven : colbert : funny
Channels // Entertainment : Humor : Short Movies

New Features
Custom Profiles
Choose your colors and interact with comments and bulletin boards
Members
New Members tab—improved search, featured members, and more!
Explore YouTube

Pretty Girls Make Graves Video Contest Winners
Join the Cybersmack Group

Recent Tags:
flyff : children : asian : car : bignasty : japanese : puke : sevenfold-beast : jpop : belife : fars : nca : hantuchova : kpop : project : monterey :

(a) Replay of YouTube with JavaScript enabled


Sign Up | Log In | Help
Broadcast Yourself
Videos
Search
Home
Videos
Channels
Groups
Members
Upload
My Videos | My Favorites | My Messages | My Subscriptions | My Playlists | My Groups | My Profile

Recently Viewed More Recently Viewed..

loading... loading... loading... loading... loading...

F-16 crash
filmed from the outside
Tags // cool
Channels // Entertainment : Humor : People
Added: 3 days ago by fatbob79
Runtime: 00:48 | Views: 16009 | Comments: 81
(116 ratings)

STANGERS WITH CANDY - Teaser Trailer - Exclusive First Look
ThinkFilm presents an Exclusive first look at the Strangers With Candy movie trailer. Keep an eye out.. the film hits theaters June 28th.
Tags // strangers : candy : amy : sedaris : david : gay : lesbian : humor : comedy : cult : classic : central : steven : colbert : funny
Channels // Entertainment : Humor : Short Movies

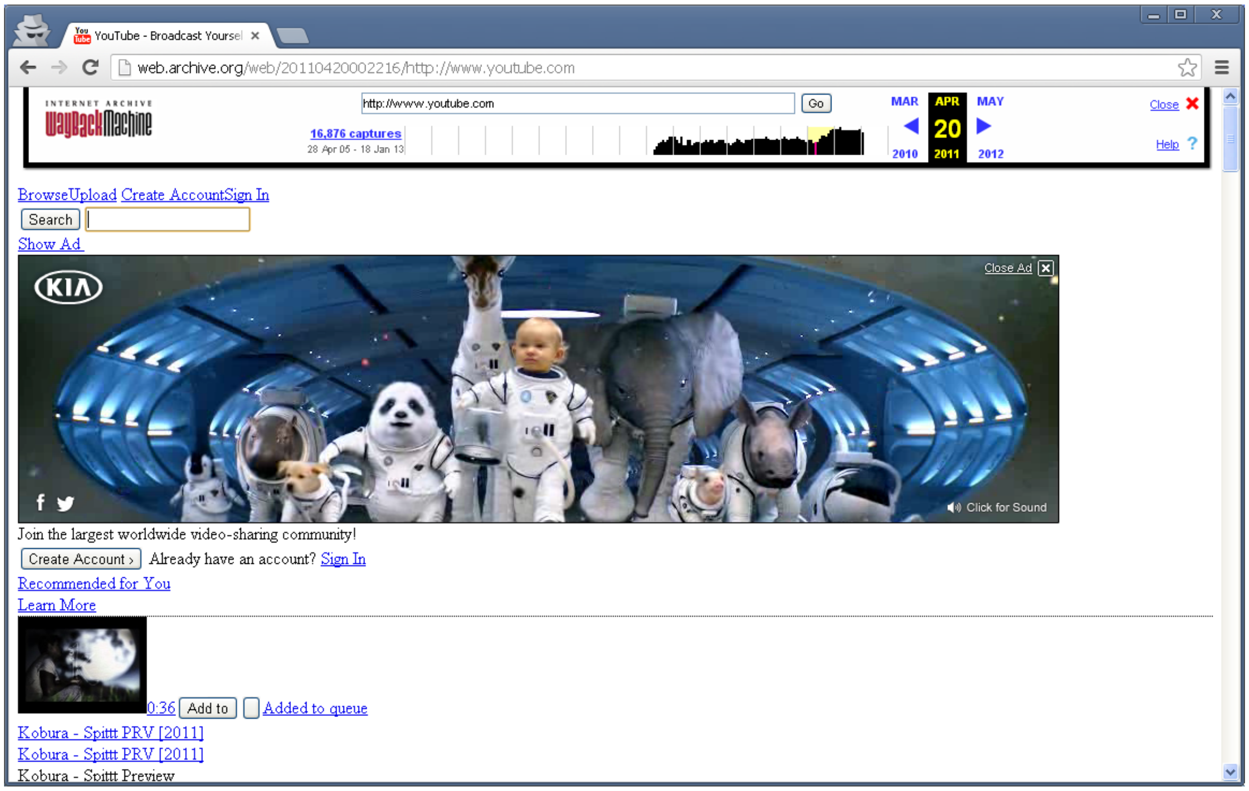
New Features
Custom Profiles
Choose your colors and interact with comments and bulletin boards
Members
New Members tab—improved search, featured members, and more!
Explore YouTube

Pretty Girls Make Graves Video Contest Winners
Join the Cybersmack Group

Recent Tags:
flyff : children : asian : car : bignasty : japanese : puke : sevenfold-beast : jpop : belife : fars : nca : hantuchova : kpop : project : monterey :

(b) Replay of YouTube with JavaScript disabled

Fig. 43 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.



BrowseUpload Create AccountSign In

Search

Show Ad

Join the largest worldwide video-sharing community!

Create Account › Already have an account? Sign In

Recommended for You

Learn More

0:36 Add to Added to queue

Kobura - Spittt PRV [2011]

Kobura - Spittt PRV [2011]

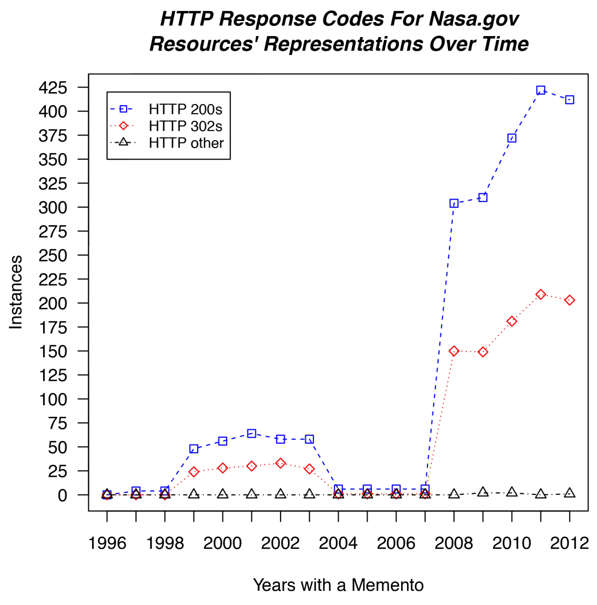
Kobura - Spittt Preview

(a)

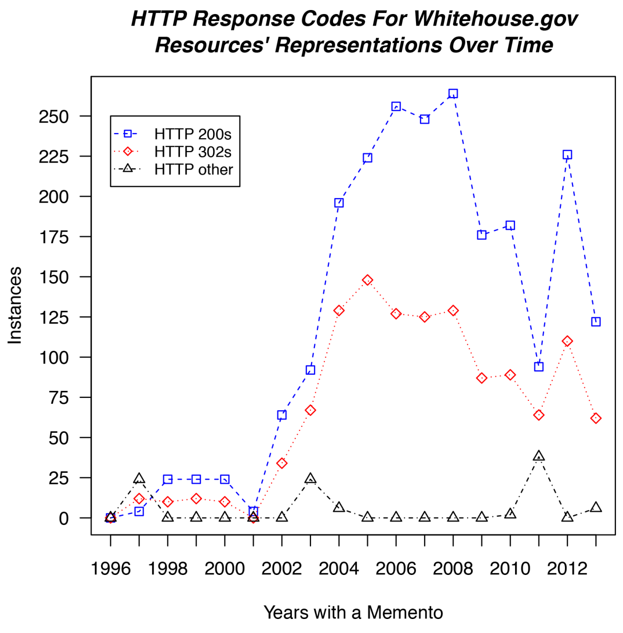
```
GET http://web.archive.org/web/20121208145112cs_/http://s.ytimg.com/yt/cssbin/www-core-vfl_OJqFG.css 404 (Not Found)
↪ www.youtube.com:15
GET http://web.archive.org/web/20121208145115js_/http://s.ytimg.com/yt/jsbin/www-core-vfl8PDcRe.js 404 (Not Found)
↪ www.youtube.com:45
Uncaught TypeError: Object #<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. 44 The 2011 capture of this YouTube.com memento (a) demonstrates the causal chain (Section 5.1) that occurs (per the browser console in (b)) when a resource is not captured.



(a)



(b)

Fig. 45 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 fewer response codes from 2004-2007 that corresponds to Figure 6 in Chapter 1 while the preservation of the White House Web page (b) exhibits a different problem yet is briefly similar in that the count drastically changed.

Table 4 Content Features of Each Collection

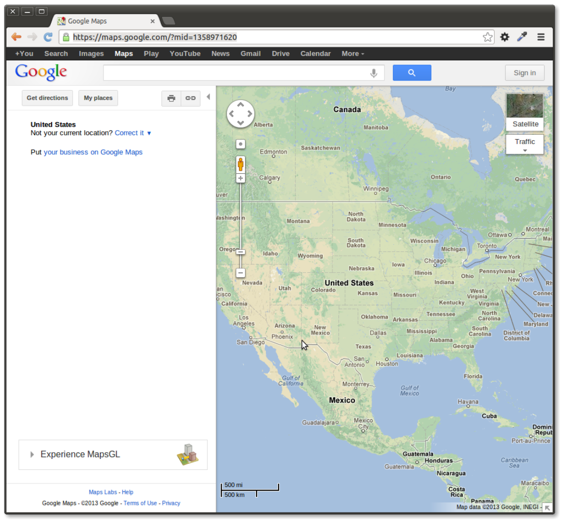
Collection	Statistical Breakdown of Content				
	PDF	Images	Other Content	HTML	HTML Content containing JavaScript
Twitter n=901	0.3%	1.3%	3.7%	84.8%	98.7%
Archive-It n=960	4.4%	0.6%	1.3%	93.7%	97.1%

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 [107].

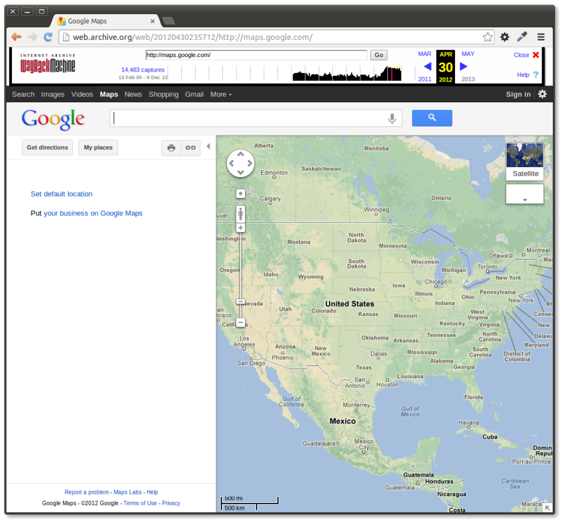
We evaluated the impact of JavaScript on archivability by using two different datasets. The first data set consisted of Bitly[1] URIs shared over Twitter. Shortened URIs are popular among social network services and the resources to which they redirect vary in expected lifespan [20]. The second data set was sampled from Archive-It (Appendix A), which was created and curated by humans. Archive-It collections often correspond to an event (e.g., National September 11 Memorial Museum) or a specific set of Web sites (e.g., City of San Francisco). The Twitter data set initially consisted of 1,000 random URIs sampled from the Twitter Garden Hose[2] and was deemed by users as important enough to share with others on social media but may have not been actively archived. From this set we removed non-HTML representations, as they do not contain embedded resources when replayed, which reduced the count to 901 URIs.

The Archive-It set consisted of the entire set of URIs belonging to the collections listed on the first page of collections on the Archive-It homepage as of October 2012. This list consisted of 2,093 human-curated URIs. From this we randomly sampled 1,000 URIs and likewise removed the non-HTML representations, which resulted in 960 URIs. As shown in

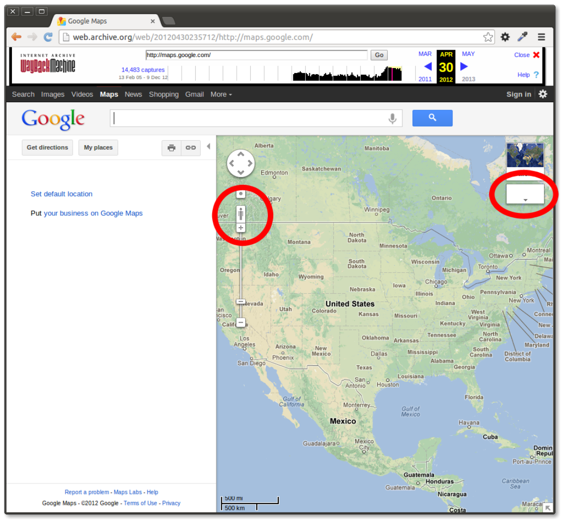
[1]`https://bit.ly`
[2]`https://dev.twitter.com/docs/streaming-apis/streams/public`


Google Maps
https://maps.google.com/?mid=1358971620
Get directions
My places
United States
Not your current location? Correct it
Put your business on Google Maps
Experience MapsGL
Satellite
Traffic
Sign in

(a) live


Google Maps
web.archive.org/web/20120430235712/http://maps.google.com/
14,483 captures
Get directions
My places
Set default location
Put your business on Google Maps
Satellite
Sign in

(b) December 2012


Google Maps
web.archive.org/web/20120430235712/http://maps.google.com/
14,483 captures
Get directions
My places
Set default location
Put your business on Google Maps
Sign in

(c) April 2012

Fig. 46 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).

Table 4, the Archive-It set has a lower proportion of non-HTML content than the Twitter set prior to removal.

We evaluated the 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 (Equation 2) then can be determined with consideration of the URI depth (number of levels down from the TLD) and F. 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 server-side parameters and client-side fragments (i.e., client-side "parameters"). The Archive-It collection has a lower $\overline{UC}$ than the Twitter collection (Figure 47), supporting the theory that the human-curated Archive-It collection deals more with higher-level URIs than the shared links of Twitter.

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

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

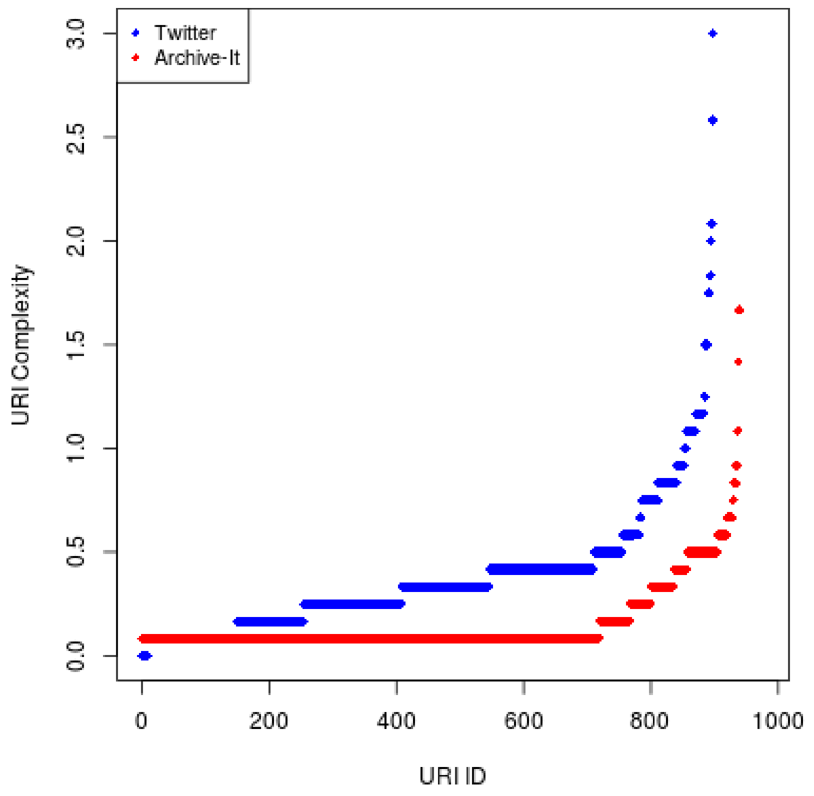


Fig. 47 URI complexity measure (UC)

To actually measure the impact on the mementos and embedded resources beyond the URIs in the collection, we established a content complexity measure CC (Equation 3), simplified for consideration of JavaScript. The Twitter set had a $\overline{CC} = 4.78$ with $CC_\sigma = 16.23$ and the Archive-It set, an average $\overline{CC} = 2.16$ with $CC_\sigma = 6.87$. 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 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 ($CC = 2.16$).

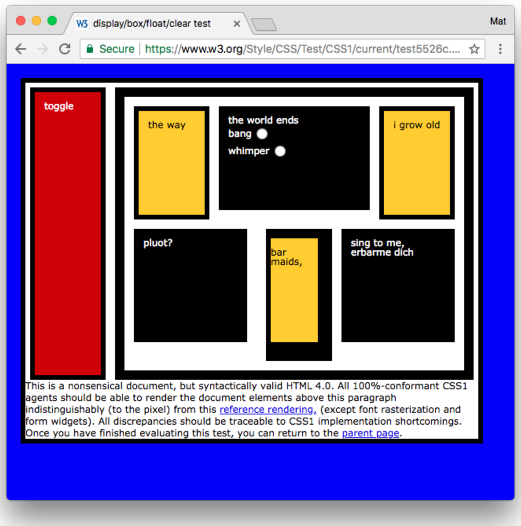
The archivability of Web sites is changing over time because of an increasing reliance on JavaScript to load resources. JavaScript is responsible for 33.2% more missing resources in 2012 then in 2005 [52] meaning JavaScript is responsible for an increasing proportion of the embedded resources unsuccessfully loaded by mementos (RQ1). JavaScript is also responsible for 52.7% of all missing content in the collections used in this study [52]. This trend is expected to increase as time progresses since the number of embedded resources loaded via JavaScript is moderately correlated to the proportion of missing content in mementos.

5.3 ARCHIVAL ACID TEST

Because archival crawlers attempt to duplicate what a user would see if they 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 (RQ2), 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 is 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.

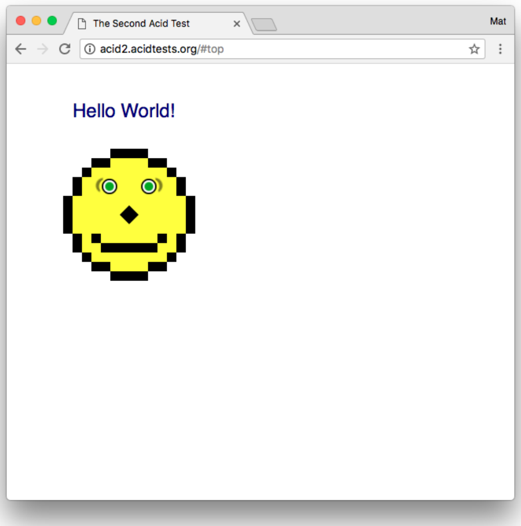
Web browsers exhibited this deviation between each other in the early days of Web Standards. A series of "Acid Tests" that implemented the Web Standards allowed each browser to visually and functionally render a Web page and produce an evaluation of how well the browser conformed to the standards (Figure 48). In much the same way, we created an "Archival Acid Test" [123] 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.

Inspired by the Acid Tests administered by the Web Standard Project (WaSP) [89], we built the Archival Acid Test [109] to evaluate how well archival tools of 2014 (when the study



This is a nonsensical document, but syntactically valid HTML 4.0. All 100%-conformant CSS1 agents should be able to render the document elements above this paragraph indistinguishably (to the pixel) from this reference rendering, (except font rasterization and form widgets). All discrepancies should be traceable to CSS1 implementation shortcomings. Once you have finished evaluating this test, you can return to the parent page.

(a) Acid1 Test



(b) Acid2 Test

Fig. 48 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 [123] to evaluate the quality of the capture of various Web archiving tools and services. A third iteration, the Acid3 Test, is displayed in Figure 50.

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 standardized features as compared to browsers,

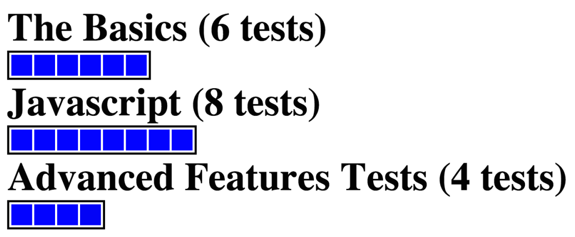


Fig. 49 The reference image for the Archival Acid Test shows what should be displayed if all tests are passed. This image represents what a user sees when viewing the test in a modern Web browser.

though they will frequently rely on a common engine utilized by browsers to stay-up-to-date.[3] 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 49) 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.

5.3.1 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://acid.example.org/1b.png`

[3]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.

1c. Remote image, absolute URI

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

1d. Inline content, encoded image

```
data:image/png;base64,iVB...
```

1e. Remote image, scheme-less URI

```
//acid.anotherserver.net/1e.png
```

1f. Recursively included CSS

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

5.3.2 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://acid.anotherserver.net/local.js" />
```

2c. Inline script, manipulates Document Object Model (DOM) tree 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)[178] 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

5.3.3 ADVANCED FEATURES TESTS (GROUP 3)

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

3a. HTML5 Canvas [201] drawing with runtime-fetched content

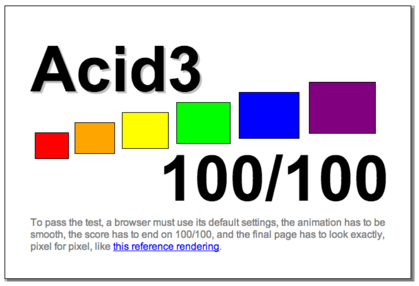
3b. Remote image stored then retrieved from HTML5 localStorage [202]

3c. Embedded content using iframe

3d. Runtime binary object

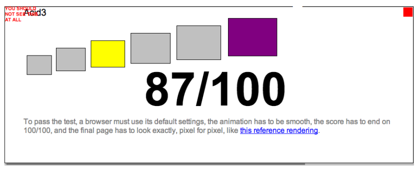
5.3.4 EVALUATION

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 50). 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.



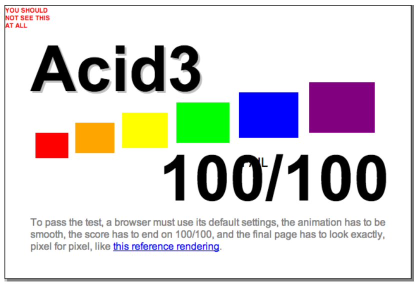
To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like this reference rendering.

(a) Chrome



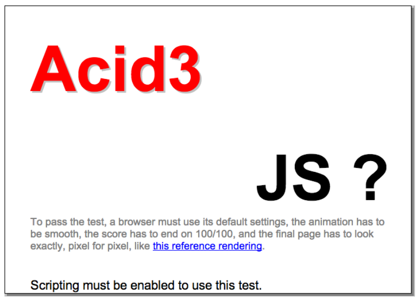
To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like this reference rendering.

(b) Archive.org



To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like this reference rendering.

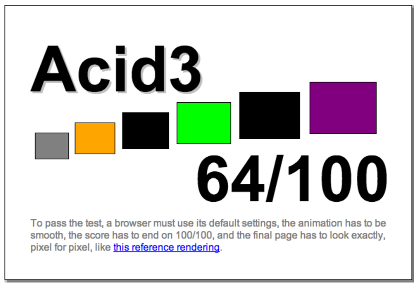
(c) Archive.is



To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like this reference rendering.

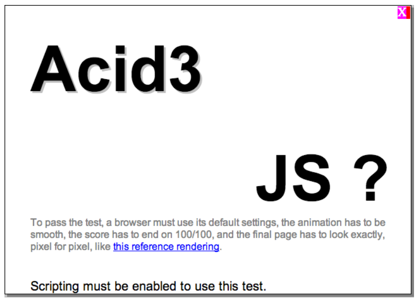
Scripting must be enabled to use this test.

(d) Mummify.it



To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like this reference rendering.

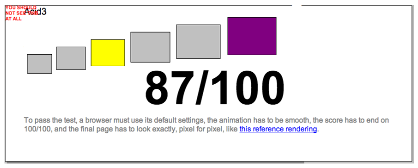
(e) Perma.cc



To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like this reference rendering.

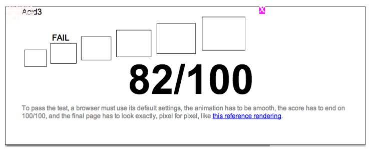
Scripting must be enabled to use this test.

(f) WebCite



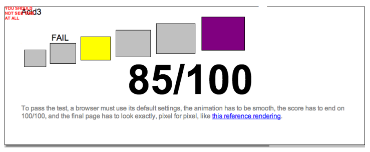
To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like this reference rendering.

(g) Heritrix



To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like this reference rendering.

(h) WARCreate



To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like this reference rendering.

(i) Wget

Fig. 50 Preliminary tests show that archival tools exhibit an incomplete feature set compared to modern Web browsers. Tests run in January 2014.

In Figure 50, we show the results of each tool's attempt at capturing the Acid3 Test Web page. Compared to the correct rendering in Chrome (Figure 50a), the five service-based tools from archive.org, archive.is, mummify.it, perma.cc, and WebCite (Figures 50b, 50c, 50d, 50e, and 50f, respectively) have more variance in their performance than the three tools of Heritrix, WARCreate, and Wget (Figures 50g, 50h, and 50i, 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.

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 5 By aligning the services' and tools' tests and failures (ca. 2014), a theme in capability (and lack thereof) is observable between the two classes.

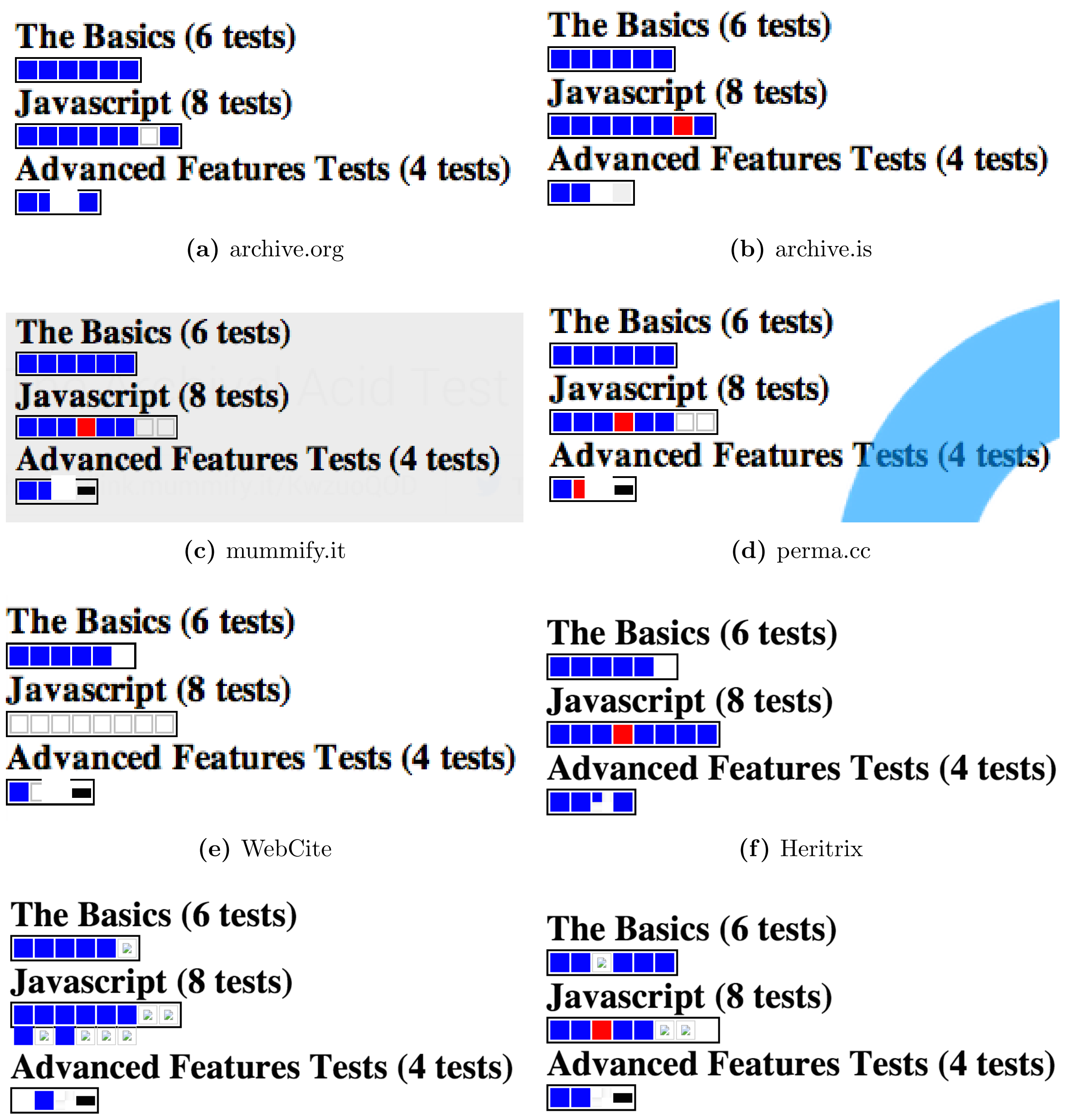
Tools' Performance

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 executed with arguments[4] including the URI 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[5].

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, and WebCite Figures 51a, 51b, 51c, 51d, and 51e, respectively) show an interesting common set of features compared to the three archiving tools (Heritrix, WARCreate, and Wget, Figure 51f, 51g, and 51h, respectively). Table 5 aligns the services' and tools' tests and failures, indicated a theme in capability (and lack thereof) between the two classes. Where archiving services exhibit a perfect record in the Group 1 set, the Group 2 set proved

[4] `wget --mirror --page-requisites --warc-file="wget.warc" http://acid.example.org`

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



(a) archive.org (b) archive.is

(c) mummify.it (d) perma.cc

(e) WebCite (f) Heritrix

(g) WARCreate (h) Wget

Fig. 51 Archiving service and tools' performance on the Archival Acid Test. Tests run in January 2014.

troublesome for all but Heritrix. Further, the nearly across-the-board failures of 2g and 3c (Figure 5) when modern browsers pass all of the tests emphasizes the functional discrepancy between archiving tools and browsers.

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 [118], we evaluated content already in the archives that existed in frames, so this discrepancy was unexpected.

Following completion of the Archival Acid Test study, additional Web archiving services and tools (e.g., Webrecorder [168] and WAIL-Electron [32]) were created that leverage a headless browser to execute JavaScript and provide a more comprehensive, “high fidelity” capture.

5.4 SUMMARY

This chapter details our studies in measuring archivability as it relates to content that is difficult to capture and replay (RQ1). We first evaluated how the capabilities of preservation tools and the technologies used on the live Web have affected the resulting archival quality (Section 5.1). After noting that archival crawlers had a difficult time preserving resources that involved dynamic fetching using JavaScript, we performed a study to evaluate the impact that JavaScript has had on archivability (Section 5.2). As JavaScript is not the sole culprit to affect archivability, we created an “archival acid test” (Section 5.3) and evaluated state-of-the-art archival crawlers and services (RQ1 and RQ2). We determined that in some cases, even the contemporary preservation tools were not creating a complete and accurate representation of live Web pages that were comprehensive of all resource representations required to display an accurate memento.

CHAPTER 6

ARCHIVAL NEGOTIATION BEYOND TIME

Of the four dimensions I could have spent my life being pushed inexorably forward through, I guess "time" isn't the worst.

- Cueball, *XKCD #1524 [146]*

Web archives preserve the live Web to represent the Web of the past. Unlike the live Web, Web archives do not similarly respond to content negotiation. However, Memento [198] provides the ability to negotiate with Web archives in the dimension of time. Because Web archives exhibit varying degrees of archival quality (Chapter 5) and (ideally) do not exhibit the ephemerality of the live Web (e.g., an HTTP 200 today should be a 200 tomorrow per Chapter 1), it would be useful to perform content negotiation on the characteristics of the resource representations held by a Web archive.

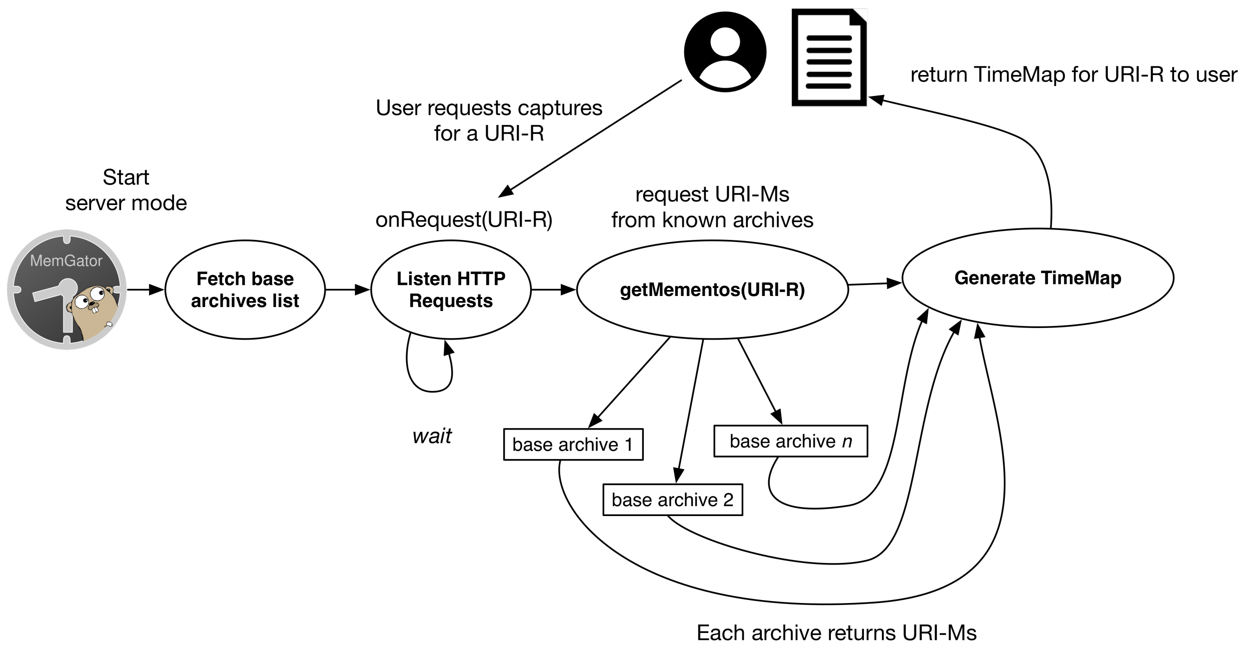
Memento TimeMaps do not currently provide standard syntax for representing additional arbitrary attributes about the mementos they describe. These attributes may be defined using a variety of methods. For example, a system for generating a TimeMap may refer to an external Web service to obtain values 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 [153] (on which the Memento Framework was based), its obsoleting successor RFC 8288 [155], and CoRE [184] syntax. In a more recent solution designed specifically with Web archives in mind, Alam et al. [6, 11] defined the CDXJ format (Section 2.4.4), an extension of the conventional CDX [99] 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 dissertation 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 mementos. 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 [51] under a certain threshold (Figures 6 and 7, respectively, from Chapter 1).

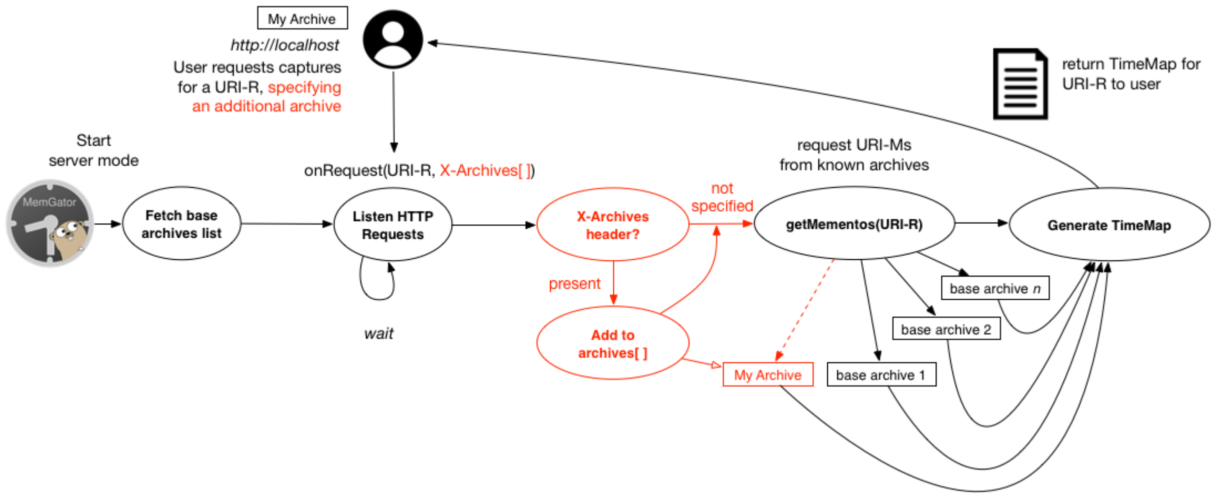
In this chapter we explore ways to resolve these outstanding issues by investigating archival negotiation beyond time. In Section 6.1 we investigate different mechanisms using existing standards for client-side specification of the set of archives aggregated by a Memento aggregator. In Section 6.2 we 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 6.3 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 will then, in Chapter 7, discuss the Mementity Framework further in the context of how it enables these additional negotiation constructs and dynamics.

6.1 CLIENT-SIDE ARCHIVAL SPECIFICATION

MemGator [10] is an open source Memento aggregator that supports CDXJ TimeMaps (Section 2.5) in addition to conventional Link and JSON formatted TimeMaps. In support of this dissertation, we adapt the code for MemGator to effectively serve as an implementation for handling 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



(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 7.2.2).

Fig. 52 (a) MemGator conventionally works on a predefined set of archives initialized on startup. By enabling clients to modify the set of archives at runtime, (b) users can effectively aggregate additional archives of their choosing through specification of an archive's attributes through an extended MemGator's HTTP endpoint.

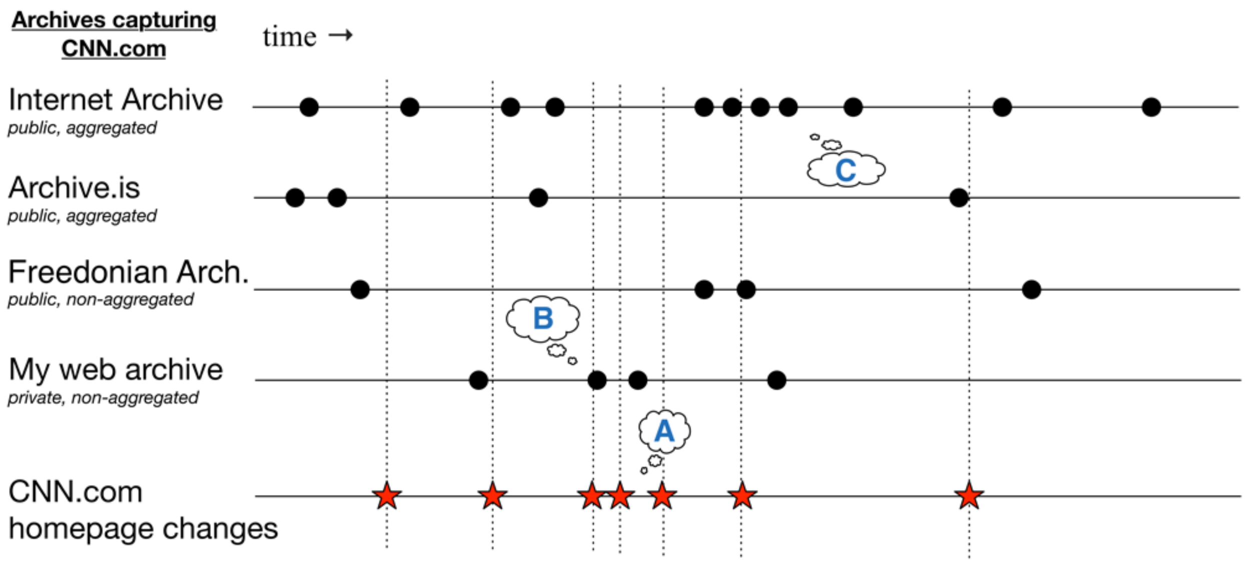


Fig. 53 Personal Web archives allow mementos from institutional archives to be supplemented. For a URI-R (e.g., `cnn.com`) that changes frequently (marker A), a Web scale archive may only preserve the page after multiple representations have occurred (marker C). Aggregation of mementos with personal 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.

server process (Figure 52a). 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 initially investigated 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` (Section 6.1.1)

2. The Prefer HTTP request header [188] with encoded JSON [45] (Section 6.1.2)

3. A client-modified, server-supplied Cookie-based [26] approach (Section 6.1.3)

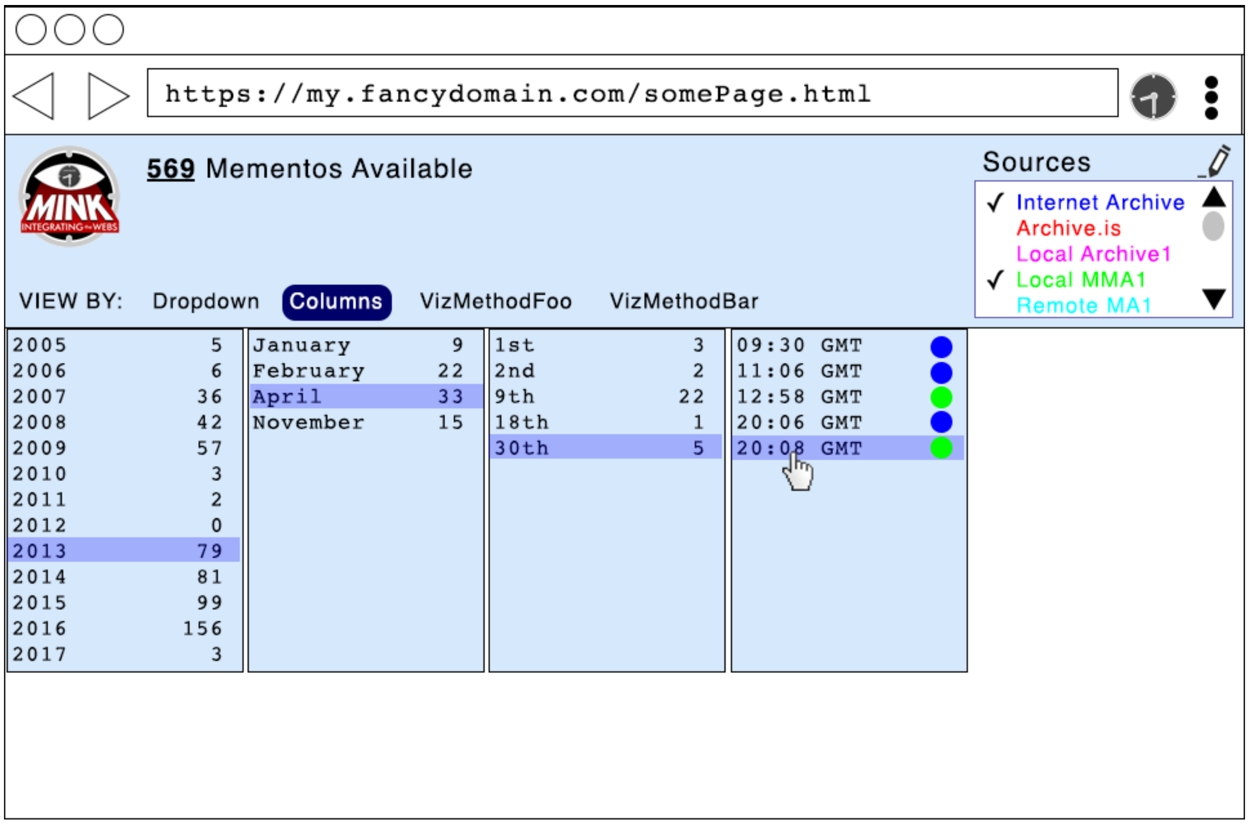
Providing the ability for a user to interact with an aggregator by providing the identifiers for archival supplementation is novel. Users often act as "pure clients" to these services in that, beyond requesting results for a URI-R, they have no say as to the sources to query for this URI-R. The purpose of allowing a client to specify a custom set of archives to an aggregator is to not necessarily to set the sources of the aggregator for all clients but rather, allow the aggregator to perform the aggregation process with a custom set of archival sources for the requesting client. 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 53). Conversely, a user may wish to aggregate mementos from a completely disjoint set of archives than queried 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 mementos (Section 1.2). The ramifications of providing private sources to public Web services are accounted for in Chapter 7. Similarly, the capability of providing a completely custom set of sources may be useful in other scenarios.

We provided a graphical means for a user to query an aggregator using their Web browser and a browser extension, Mink (Section 4.4). The initial capability of Mink [122] did not allow for the user to customize the set of archives aggregated but parsed, interpreted, and displayed results (e.g., memento count, available datetimes) as returned from a Memento aggregator relative to the URI being viewed in the browser. We subsequently created a mockup of how we anticipated the interface for client-side archival specification from Mink (Figure 54). The interface to accomplish this required features of the framework (Chapter 7) to be discussed inclusive of query precedence and short-circuiting (Section 7.2.2) and hierarchical interoperability with the formats returned from both Web archives and Memento aggregators alike (Section 7.1.1). Additional design considerations for Mink to realize the necessary capability through a user interface are described in Section 8.3.4. The following subsections describe the approaches at accomplishing client-side archival specification at a lower level than exposed to the client in Mink.

6.1.1 SPECIFICATION USING X-ARCHIVES

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 7.2.2). In a preliminary prototype[1], 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 (Figure 52b). 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

[1]https://github.com/machawk1/gogator/



2005	5
2006	6
2007	36
2008	42
2009	57
2010	3
2011	2
2012	0
2013	79
2014	81
2015	99
2016	156
2017	3

January	9
February	22
April	33
November	15

1st	3
2nd	2
9th	22
18th	1
30th	5

09:30 GMT	●
11:06 GMT	●
12:58 GMT	●
20:06 GMT	●
20:08 GMT	●

Fig. 54 Mink initially communicated with a Memento aggregator to parse and display results of the request in the browser for further navigation. This initial mockup of Mink would introduce the ability for clients to specify the set of archives aggregated and exhibit features of the framework. However, unless Mink is itself performing the aggregation, the aggregator must understand the semantics and syntax of client-side archival specification.

disjoint set. Despite the naïve and ad hoc approach (and the deprecation of "X-" prefixed headers [181]), 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 neither scalable, nor semantically expressive, nor standard, so we opted to investigate more standards-driven method for client-side archival specification.

6.1.2 SPECIFICATION USING PREFER

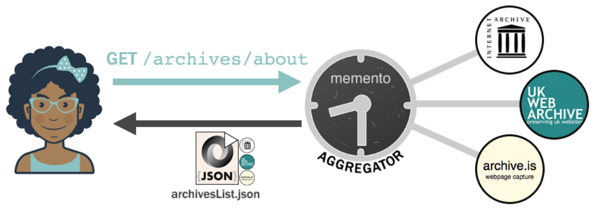
A second approach we investigated is to allow client-side archival specification using HTTP Prefer (Section 2.3). Specifying preference first requires knowing the expected format of expression by the server then modifying a supplied example of this format or generating one that adheres to the format. Prefer is similar to the Expect header [79] with the exception that servers are allowed to ignore the stated preference [188]. Figure 55 shows an example where a user requests the list of archives from an aggregator (Figure 55a) and receives a JSON payload containing three archives (Figure 55c). Interacting with aggregators in this manner requires a more capable and transparent aggregator than is currently exhibited by conventional Memento aggregators (and is addressed in Section 7.1.1). After receiving the list of supported archives ($\{A_0\}$), a client may use the Prefer header to construct their own list of archives, matching the format (e.g., JSON) that the aggregator specified. This approach of first querying the aggregator without the typically supplied URI-R parameter 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 Prefer specification does not allow line breaks (often present in JSON) within the Prefer header field, like many other HTTP headers, so for the field to contain a JSON value as a preference, it can be encoded prior to transmission. 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, for example:

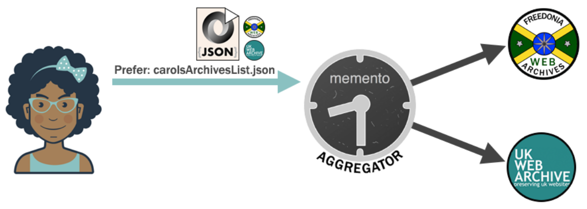
```
Prefer:  archives="data:application/json;charset=utf-8;base64,Ww0KI...NCn0="
```

Alternatively, as an example and based on the expected format returned from an aggregator, preference communication might also be expressed differently by a client, for example:

```
Prefer:  archives="data:text/x-yaml;charset=utf-8;base58,2bDfoZJb3kKr...WbWap"
```



(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. 55 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.

Figure 55 shows Carol using the format of archival specification provided by the aggregator to construct a JSON file containing the specification of two archives (Figure 55d) and submitting this back to the aggregator (Figure 55b) to be applied onto subsequent requests. Alternatively, a URI of a remote JSON file may also be supplied using standard Prefer syntax, for example:

```
Prefer: archives="https://git.io/archives"
```

This latter approach could be useful for a user to post and share configurations by-reference, however, it limits the ability for client-side manipulation. Because of this, we focus on the explicit (by-value) specification of the configuration as with the former example to allow for manipulation of the list of preferred archives at the time of request. We also considered using extensible syntax like

```
Prefer:  config={archives:[...]}
```

instead of

```
Prefer:  archives=...
```

(where both colored preference values would be encoded) but opted for the latter to reduce the potential of a semantic clash for usage of Prefer from other domains beyond our archive-related use cases (see Section 9.2)

More sophisticated aggregation may require filtering on a memento-level (e.g., only source mementos from archives with a certain quality) or on a TimeMap-level. Memento TimeGates 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 55b) 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 55a).

2. The aggregator returns the set of archives, e.g., as a JSON (per MemGator) or an XML (per `mementoweb.org`) file (Figure 55a), represented as $\{A_0\}$.

```
> 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. 56 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.

3. Once a response is received from the aggregator (e.g., Appendix B), 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 55b) and submit back to the aggregator for subsequent queries (Figure 56).

While a client may repeatedly provide this archival specification with each request, adapting existing practice as exhibited on the live Web (e.g., Cookies [26]) might allow for persistence of preference. Existing means of expressing persistence of preference could serve as a way for customization of a specification received through manipulation after receiving the response. For instance, a profiling probability [12] may be manipulated or a value of query precedence (Section 7.2.2) may be modified.

Prior to the work in this dissertation, no Memento aggregator currently supports client-side archival specification. Both MemGator and Webrecorder's aggregator use JSON for internal archival specification, i.e., which archives should be queried as pre-configured on the server. Because of this, we adapt the server-side notions in these implementations with the assumption going forward that a JSON response will be received from a more capable aggregator. A client may perform step 3 (above) using the HTTP Prefer request header. After potentially manipulating the JSON response following receipt, 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 56).

An aggregator has the option of complying with the requested preference fully, partially, or not at all. This level of fulfillment of the expressed preference is communicated through the HTTP `Preference-Applied` response header from the server (aggregator). In much the same way that the JSON representation of the client's preferred set of archives is encoded prior to being sent in the request from the client, the returned `Preference-Applied` header can be decoded for verification by the client. An identically encoded string, as sent in the request within the response, may be an indication of comprehensive fulfillment of the requested preference by the server. However, a preference may be comprehensively fulfilled yet the JSON supplemented with additional attributes by the aggregator for each archive (Figure 60). For example, an aggregator may add "probability" attributes to each JSON object to align with the semantics in MemGator's archive specification. As another example, an aggregator may also associate keys to each archive (Figure 61) to allow for a level brevity via symbolism within subsequent requests by the client. This might allow clients to query the same set of archives using the aggregator much more succinctly.

```
[
  {"timemap": "http://archive.alice.me/web/timemap/link/"},
  {"timemap": "http://arch.carol.org/web/timemap/link/"},
  {"timemap": "http://bob.net/archive/timemap/cdxj/", "id": "bob"}
]
```

Fig. 57 Bob sends a request (Figure 58) with an archival specification describing three archives' TimeMap endpoints. Encoding this JSON prior to sending via Prefer produces a string (abbreviated here) of `WwogIHs...QpdCg==`.

```
> GET /timemap/cdxj/https://matkelly.com
> Host: aggregator.example.com
> Prefer:
 ↪  archives="data:application/json;charset=utf-8;base64,WwogIHs...QpdCg=="
```

Fig. 58 A client encodes the JSON (Figure 57) and includes it with the request to an aggregator.

As an example to illustrate the prior points, Bob has queried an aggregator capable of client-specified aggregation as Carol did in Figure 55a to find that this aggregator expects configurations to be in the typical JSON format. He discards what is returned from the aggregator (e.g., Figure 55c) and constructs his own JSON describing TimeMap endpoints for archives of his choosing (Figure 57). He (or a software tool to automated the process) base64 encodes the JSON (to "`WwogIHs...QpdCg==`") and submits a request to the aggregator containing this encoded string in the Prefer header to the aggregator (Figure 58). The aggregator may comply with his request and perform the aggregation from the archives as specified. The response from the aggregator with the same hash in the `Preference-Applied` header (Figure 59) may indicate this, though Bob may first wish to decode this string to ensure that all of his preferences were considered and applied. In a second scenario, the aggregator has responded with a different hash in the `Preference-Applied` header (Figure 60). Bob (or again, his automated tool) decodes this value ("`WwogIHs...n0KXQoK`") to find that the JSON includes what he specified with additional information attributed to each JSON object describing an archive. Figure 61 shows this addition with the aggregator having added `id` attributes to each archive as well as added additional TimeGate endpoints for two of the three archive entries.

```
< HTTP/1.1 200 OK
< Preference-Applied:
 ↪  archives="data:application/json;charset=utf-8;base64,WwogIHs...QpdCg=="
```

Fig. 59 A server replying that the applied preference has the same hash might be indicative that the preference was fully applied as requested.

```
< HTTP/1.1 200 OK
< Preference-Applied: archives="data:application/json;charset=utf-8;base64,
 ↪  WwogIHs...n0KXQoK"
```

Fig. 60 An aggregator returns a `Preference-Applied` response header with a different value.

```
[
  {
    "id": "alice",
    "timemap": "http://archive.alice.me/web/timemap/link/",
    "timegate": "http://archive.alice.me/web/timegate/link/"
  },
  {
    "id": "carol",
    "timemap": "http://arch.carol.org/web/timemap/link/",
    "timegate": "http://arch.carol.org/web/timegate/link/"
  },
  {
    "id": "bob",
    "timemap": "http://bob.net/archive/timemap/cdxj/"
  }
]
```

Fig. 61 The server's response, when the string `WwogIHs...n0KXQoK` (Figure 60) is decoded, shows that additional TimeGate endpoints were added, attributes for each archive shuffled (order of attributes is not significant in JSON objects), and additional attributes added to some of the supplied archives.

Prefer appears to be a suitable means to express client preference and has been discussed and minimally realized elsewhere. External relevant uses of Prefer are in a second RFC's [147] extension of keyword and clarification of semantics as relevant to WebDAV. Jones [104] and Van de Sompel [197] (Section 3.1.3) each described using Prefer for specifying whether mementos should be rewritten. Rosenthal [173] expanded on some potentially useful preferences like `banner-inserted` and `url-rewritten`. pywb [129] also began supporting Prefer in the style of Jones and Van de Sompel with the addition of the specification of whether the archival banner should be displayed. None of this prior work addressed client-side archival aggregation but mainly focused on using Prefer to affect the representation of an individual memento.

6.1.3 SPECIFICATION USING COOKIES

HTTP Cookies [26] provide a native, familiar, transparent interface for Web users, particularly those that use Web browsers, to maintain preference between sessions. A typical usage of Cookies on the live Web is to maintain a key for persistent login or to store a session identifier on the client-side to allow a preference to persist. A server may specify a `Set-Cookies` response header to instruct a client to create a Cookie with a set of key-value pairs representing metadata. Additional metadata about the Cookies, like the applicable `Path` and `Domain`, may also be specified by the server. A user agent may ignore the `Set-Cookies` header in its entirety [26].

Cookies persist when a Web browser is closed or a system rebooted. While Cookies were originally also a means for client-side storage of a small amount of information, there are more modern APIs like Web Storage [202] (e.g., localStorage, sessionStorage) and IndexedDB [5]. Once set, a user-agent will send the Cookie with each subsequent request that meets the conditions defined in the Cookie (e.g., the Cookie Domain or Path). This process is often transparent to the user.

Cookies do not have any form of confirmation that an attribute customized by a user on the client was accepted by the server. With requests to conventional Memento aggregators, the absence of URI-Ms from an archive for a requested URI-R is often an implicit indication that the archive in question contains no mementos of this URI-R. Explicitness of the set of archives aggregated is a goal of this work to further integrate the client in requests to archives using aggregators. Because Cookies have no means of acknowledging that the configuration that was supplied by a client has been considered, they are insufficient to meet this explicitness requirement. Prefer, on the other hand, contains a semantic and syntactic mechanism to express exactly this through the `Preference-Applied` HTTP response header. Because of this lack of explicitness of Cookies, the non-standards basis of `X-Archives`, and the suitability of Prefer to what the framework attempts to accomplish, we opted for the solution involving Prefer.

6.2 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 as to how aggregators can indicate content that requires special handling when dereferenced for both replay and Memento-style aggregation (RQ4). This section details the enrichment of TimeMaps to produce StarMaps.

6.2.1 CONTENT-BASED ATTRIBUTES

Determining how many mementos exist from an archive for a URI-R is impossible from a TimeMap alone [115, 116]. 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 [79], `content-type` [79], and `Last-Modified` [77], are often used to gain information about archival holdings without requiring each URI-M be repeatedly dereferenced. 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 others do not.

6.2.2 DERIVED ATTRIBUTES

Researchers often analyze the contents of a memento and generate derived data from this analysis. For example, Brunelle et al. [50, 51] 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 [113, 7], we explored using content addressing to facilitate deduplication 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 [68]), 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 [16] applied the SimHash [56] algorithm with $k = 4$, 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-M. Retaining these values and expressing them in TimeMaps for a URI-M would save users of TimeMaps from having to regenerate data based on these and other of derived attributes.

6.2.3 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 (Figure 65) 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 62).

Standard authentication procedures and access patterns as used on the live Web helped 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

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

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 URI for authentication. After authentication using a similar method to the live Web, the user can use the token to access private Web archive captures as configured by the authentication server (Section 7.1.2) 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 allows persistent access using a token derived from authentication. 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 63 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 second location to obtain this. Upon obtaining the credentials from the user, a token is returned that is 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.

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 64 shows an example enriched CDXJ record containing attributes describing how the token is stored in an enriched CDXJ StarMap. The example uses OAuth2 [86] for authorization when dereferencing URI-Ms with

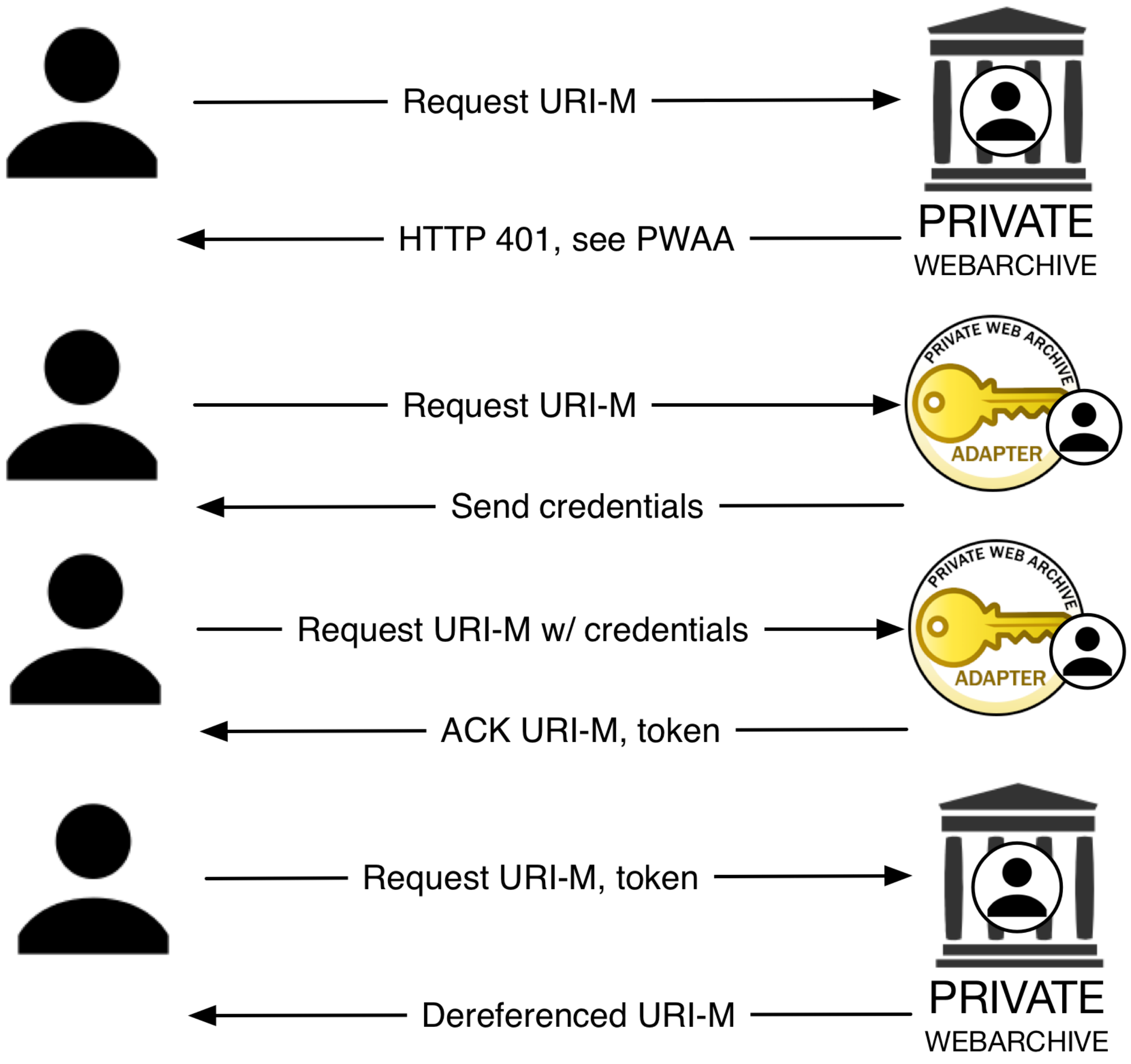


Fig. 63 A private Web archive may deny anonymous access to its contents, potentially reporting an HTTP 401 even if it contains no captures for a URI-R. The archive should then refer the client to a Private Web Archive Adapter to authenticate and obtain a token that can then be used to request the contents of the private Web archive.

```
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": "c6ed419e74907d220c69858614d8669ff3732df0cc5647ef0a3..."
  }
}
```

Fig. 64 An amended CDXJ record for a private capture of `facebook.com`. Line breaks added for readability.

this field and the BLAKE2 hashing algorithm [180] for tokenization for persistent access to private mementos.

6.3 SOURCES OF DERIVED ATTRIBUTES

CDXJ allows metadata fields (lines beginning with `@meta`) about the TimeMap to precede the listing of captures. Figure 65 contains metadata fields (highlighted in red) within a CDXJ TimeMap that are typically also found in a Link-formatted TimeMap (Figure 66), e.g., URI-R for the original resource, TimeGates, other related TimeMaps, etc. With the introduction of derived attributes (Section 6.2.2), 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.

Figure 62 provides an example where a derived attribute requiring calculation (memento damage [51]) 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 [85]), for instance, instructs parsers to look to the URI specified in the template

```
!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. 65 An abbreviated CDXJ TimeMap from MemGator for `facebook.com`. CDXJ 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. 66 An abbreviated Link TimeMap from MemGator for `facebook.com`. The portions colorized in red correspond to the metadata records of the CDXJ TimeMap in Figure 65.

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" [157] wherein the expectation is for parser to look to the URI-M where the access attribute exists for resolution. The "via" `rel` value [157] 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.

6.4 CONTRIBUTIONS TO RESEARCH QUESTION 4

Users most often interact with Web archives in the dimension of time, as enabled by Memento [198]. In this chapter, we provided a means for content that was captured behind authentication to signal that it requires special handling (RQ4). This is enabled by first extending on Memento concepts of TimeMaps with our introduced StarMaps that allow for expression of dimensions beyond time for mementos. We defined three classes of attributes (Section 6.2) that we initially anticipate being useful to express with examples of each. The third type, Access Attributes as described in Section 6.2.3, provides the mechanism to answer RQ4 while the other two attribute types have additional use cases facilitated by the work in this dissertation. We provide sample mechanism to define these attributes in Section 6.3.

In introducing a mechanism for private captures to signal privacy, we also enabled clients to have more control over the sources of archives that are used by aggregators (Section 6.1). With this power of interaction with archives, a client can control the sources used, inclusive of ones they control containing their private and personal captures, as well as systematically regulate access to these captures at the time of aggregation. The dynamics of the latter that utilize the concepts introduced in this chapter are described in detail in Chapter 7.

CHAPTER 7

A FRAMEWORK FOR AGGREGATING PRIVATE AND PUBLIC WEB ARCHIVES

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 [191]*

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 our previous work described thus far in this dissertation. This chapter addresses work 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 provides 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 dissertation we introduce three mementities: the Memento Meta-Aggregator (Section 7.1.1), the Private Web Archive Adapter (Section 7.1.2), and the StarGate (Section 7.1.3).

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. 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 7.1.1). 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) 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 [10], 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 Chapter 6, we outlined negotiation of this sort in the context of how to express these additional dimensions in the conventional TimeMap medium 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 7.1. 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 7.2.2), and potential for inter-archive and archive-to-user collaboration, as discussed in Section 7.2. Section 7.3 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 7.4 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.

In this chapter we will outline our research [110, 124] toward a framework to aggregate public and private Web archives.

7.1 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

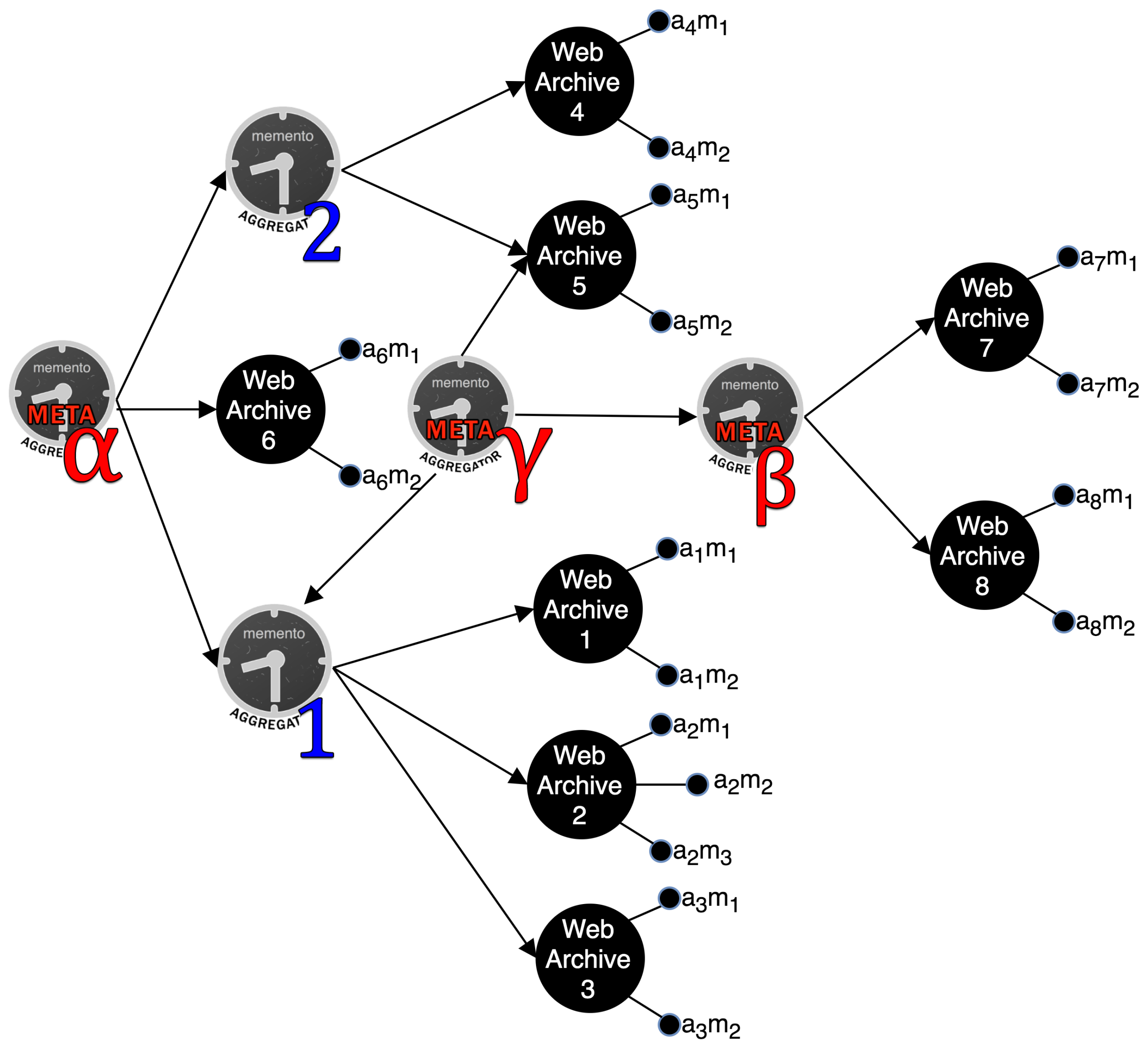
Reference implementations for each mementity will be provided as software to serve the respective role of their purpose in the Mementity Framework. Extensive details about each mementity are provided in Sections 7.1.1, 7.1.2, and 7.1.3, respectively.

7.1.1 MEMENTO META-AGGREGATOR

Memento Aggregators 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 7.1.2). A conventional MA is not required to be present for an MMA to function. 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.

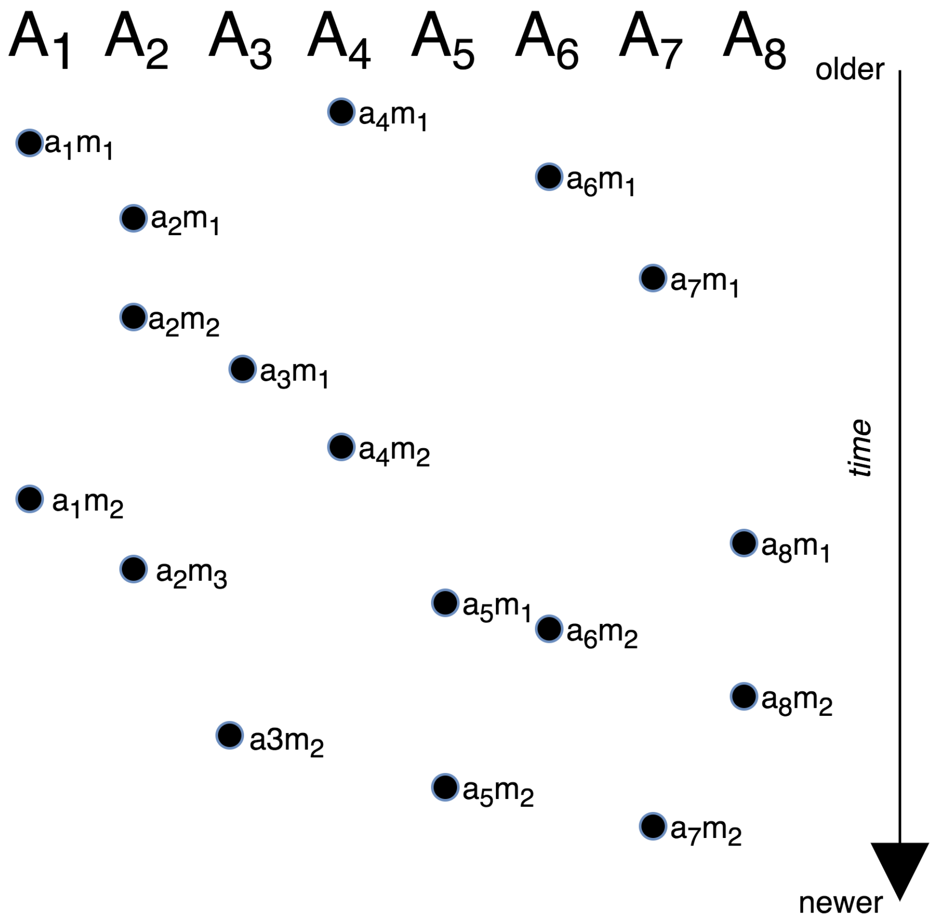
Figure 67 describes a sample hierarchical relationship of mementities consisting of MMAs, MAs, and Web archives (WAs). When MA$_1$ receives a request for URI-Ms for a URI-R, for instance, the request is relayed to WA$_1$, WA$_2$, and WA$_3$ for the sets of mementos $\{a_1m_1, a_1m_2\}$, $\{a_2m_1, a_2m_2, a_2m_3\}$, 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 68a. The ordering for the mementos contained within the configured archives as requested from various mementities is displayed in Figure 68b. This figure also shows examples of an MMA obtaining results from multiple MAs (e.g., MMA$_\alpha$ from MA$_1$ and MA$_2$) and even MMAs referring to other MMAs for their results when queried (e.g., MMA$_\gamma$ referring to MA$_1$, WA$_5$, and MMA$_\beta$ with the latter referring to WA$_7$ and WA$_8$). The configuration of MMA$_\beta$ is similar to the relationship of MMA$_{Carol}$ to MMA$_{Alice}$ in Figure 69 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 be 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.



$A_{1\ldots n}$	Archive 1 of n
$MA_{1\ldots n}$	Memento Aggregator 1 of n (denoted using integers)
$MMA_{\alpha\ldots\omega}$	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. 67 Memento Meta-Aggregators may aggregate URI-Ms from multiple 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 68.



(a) Temporal ordering of mementos aggregated in the hierarchy described in Figure 67.

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. 68 The temporal ordering of URI-Ms in a StarMap depends on the set of archives aggregated in a StarMap. Per Figure 67, 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 [125], and replays them using her local Wayback instance within WAIL [120] (Section 4). 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 in Chapter 1)[1]. 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 69 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 instance is configured with every request from the client. Furthermore, the set of archives to which the request is relayed

[1]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 7.1.2.

is static as was configured when initializing the service [114]. Aside from the dynamics of how a client specifies which archives to aggregate at the time of request (Section 6.1), 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 54 in Chapter 6 showed 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 6.1 in anticipation that the endpoint (currently a running MemGator instance) understands how to interpret the archival selections. The implemented deliverables and design decisions as adapted to the constraints of a browser extension are discussed in Chapter 8.

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. This new capability enables a more user-friendly method of configuring an aggregator and is novel in that all existing implementations of Memento aggregator are server-side mementities that a client queries. As relevant to our work [7, 113, 119] on integrating Web archives with IPFS (Section 4.5), Mink can also leverage the JavaScript port of IPFS [163] to allow client-side browser-based instances of Mink to communicate with others' Mink instances. With the additional capabilities in Mink, the potential for aggregation to be both collaborative as well as purely client-based would allow for further exploration beyond this dissertation; however, our initial implementation is described in Section 8.3.4. A more conventional, still purely client-based use case is described in collaboration of Web archives using ipwb in Section 7.2.4.

7.1.2 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

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`

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

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

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

Fig. 69 Three Memento Meta-Aggregators are configured to perform selective aggregation.

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 64 (in Chapter 6) 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.

Figure 70 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.

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. 70 Abstraction of the authentication to private Web archives follows a flow similar to OAuth 2.

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, a token may be represented within StarMaps sourced from an MMA or a Web archive that directly provides StarMaps independent of aggregation. 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 (Chapter 6). 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, supplementing TimeMaps with additional attributes for mementos may be used to consider archival content negotiation in other dimensions. The mementity in the next section takes these concerns into consideration.

7.1.3 STARGATE

Memento TimeGates generally accept a URI-R and a datetime (through the Accept-Datetime HTTP header [198]) 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 7.1.1 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 the parametric exclusion approach of, "All archives for which you are configured except for Bob's and the Internet Archive".

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 [116, 115] 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 non-relevant URI-Ms (e.g., `facebook.com` login pages) and prevent misrepresentation of the quantity of captures for a URI-R [116].

7.2 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 7.2.1), a precedence model for advanced querying of archives for aggregation (Section 7.2.2), client-side specification of archival selection (Section 7.2.3), and collaboration and propagation of Web archives beyond the perspectives of the archives themselves (e.g., between peers, Section 7.2.4).

7.2.1 NEGOTIATION APPROACHES

Our previous work [117] 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 [194, 106] have created implementations to solely interact with the memento in the original medium of the mobile Web.

```
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. 71 A user requesting a StarMap from a StarGate where damage of all URI-Ms is less than 0.5.

In Section 2.2 we discussed the Prefer HTTP header [188], which provides a basis for content negotiation in other dimensions. In Section 6.1 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 [188]. Though the specification consists of a registry of preferences (e.g., `return=minimal` and `return=representation`), Van de Sompel et al. [197] 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 [104], unmodified content from a Web archive instead of content that is rewritten by the archival replay system.

Figure 71 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 [188]. 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.

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 71 prior to the StarGate's completion may return a `102 - Processing` status [82]. When a preference has been applied to a requested StarMap from a client to a StarGate, the response will contain the

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

Fig. 72 Upon completion of the potentially temporally expensive procedure of calculating damage for all URI-Ms for `http://facebook.com` from Figure 71, a StarGate will respond with a header containing the applied preference.

`Preference-Applied` HTTP response header [188] and an HTTP 200 (Figure 72). This preference is propagated to the list of metadata headers in a CDXJ StarMap, similar to those highlighted in red in Figure 65 (Chapter 6). An example of this procedure is illustrated in Appendix F.

7.2.2 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 73 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 [133] example in Section 3.4.2, access to this content as an act of 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.

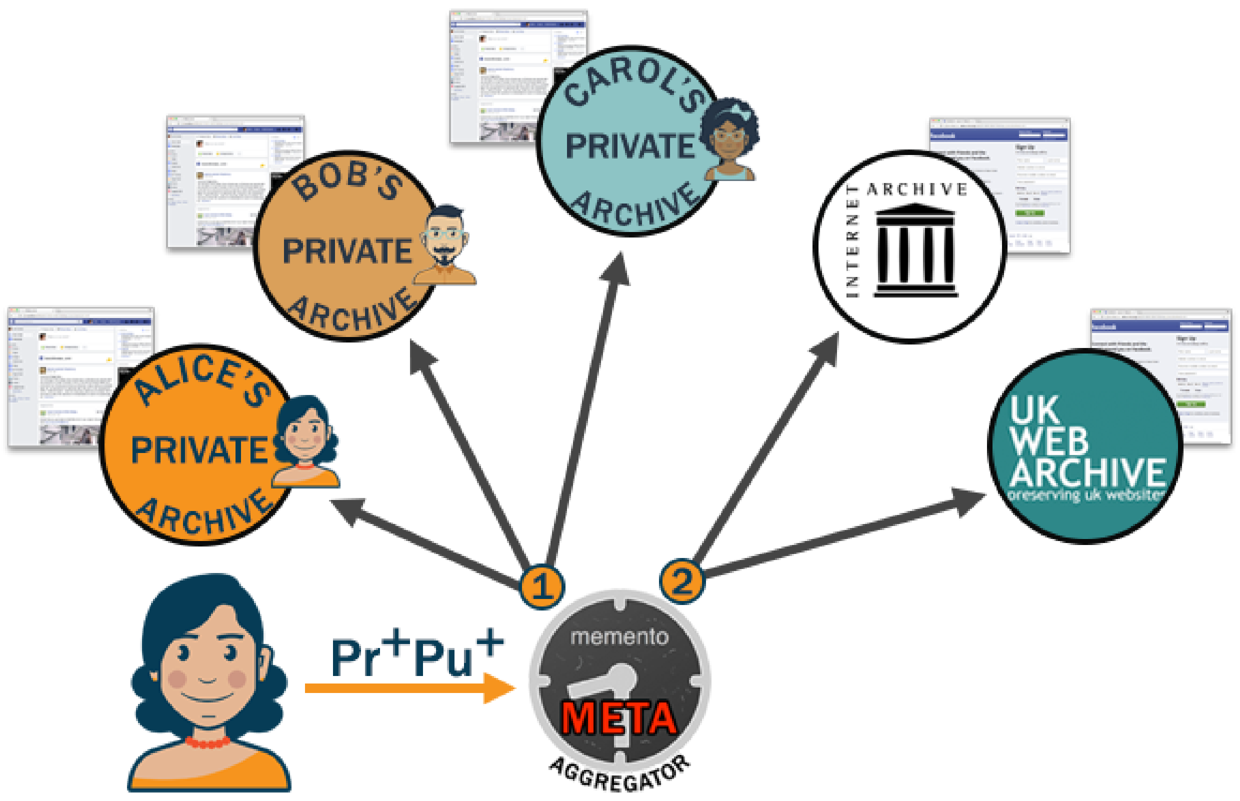


Fig. 73 Archival precedence using private first then public Web archiving querying model (Pr^+Pu^+).

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 identified three methods for identifying personalized representations [117]. Of the methods proposed, we did not investigate (we opted for one of the other three) specifying 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 73). 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 in Chapter 1), the latter is less useful in observing how the page has changed over time. To maintain relevancy of the desired sort of representation, we 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.

In the basic model below, we express various access precedence models (henceforth *profiles*) 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.

$$noArchives \rightarrow \varnothing \rightarrow \{\} \tag{4}$$

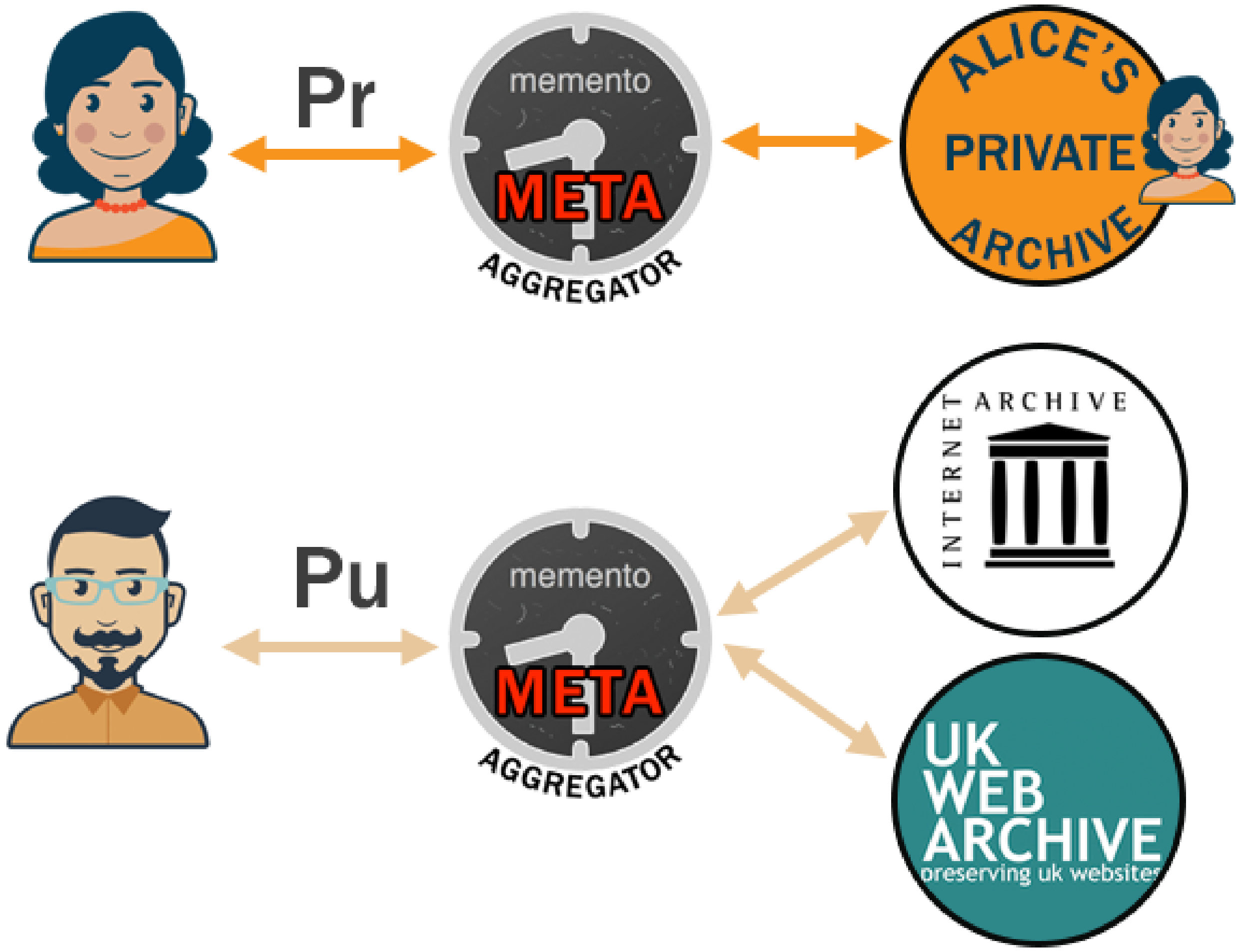


Fig. 74 PrivateOnly (Pr) and PublicOnly (Pu) aggregation in an MMA.

$$publicOnly \rightarrow {P_u}^+ \tag{5}$$

$$privateOnly \rightarrow {P_r}^+ \tag{6}$$

$$privateFirst \rightarrow {P_r}^+{P_u}^+ \tag{7}$$

$$publicFirst \rightarrow {P_u}^+{P_r}^+ \tag{8}$$

The basic profiles pair with the syntax of the `profile` relation type [205], allowing clients to request resulting TimeMaps containing URI-Ms from a subset of archives from which the Memento mementity requests (Figure 74). 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 7) 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 from Chapter 3). We discuss this usage and access pattern further in Section 7.3.5.

7.2.3 MMA ARCHIVE SELECTION

Here we revisit the scenario introduced in Section 7.1.1, and abstracted in Figure 69 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 7.1.2). 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 69 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 it 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.

7.2.4 COLLABORATION AND PROPAGATION

Collaboration in Web archives is often exhibited by individuals and organizations submitting URIs to a centralized 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 dissertation we focus on collaboration of Web archives by-value (sharing WARCs) and distributed (i.e., non-centralized) collaboration by-reference. In Section 4.5 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 7.1, particularly the Memento Meta-Aggregator from Section 7.1.1, 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 4.5). Our initial approach at privacy in ipwb entailed performing symmetric encryption to the content prior to disseminating it into IPFS [119]. 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 75 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.

In Section 7.2.3 we briefly discussed the capability of Mink instances exhibiting the capabilities of MMAs communicating with one another for peer-to-peer, purely client-driven archival querying and aggregation. In related work leveraging state-of-the-art technologies for Web archiving, we leveraged a then-young WebRTC protocol to facilitate the replication of NASA satellite imagery posted to the Web [111]. In addition, we [8] 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

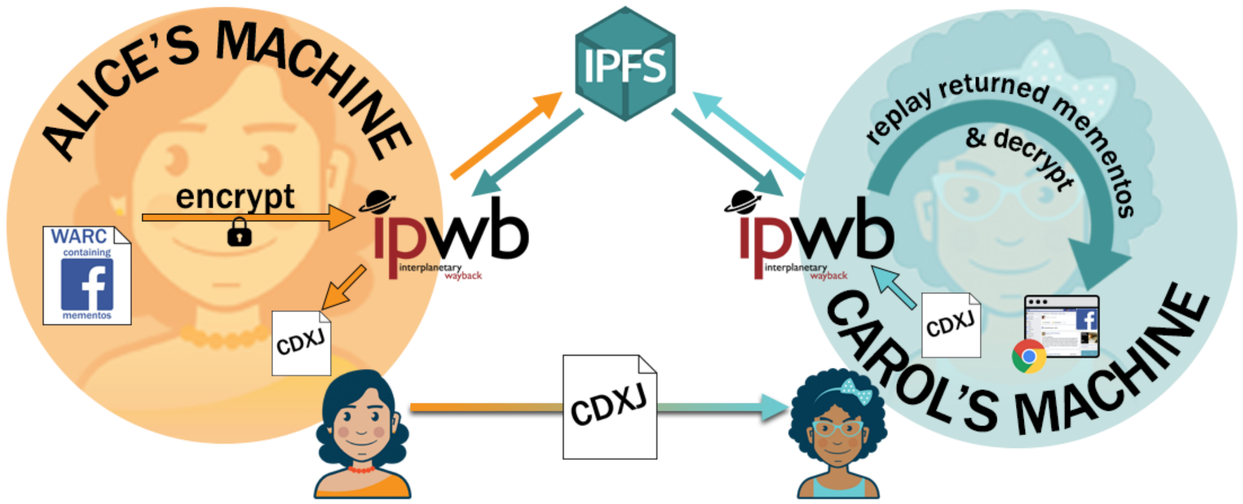


Fig. 75 The extended ipwb model for collaboration involves symmetric encryption and decryption of the payload prior to dissemination. When Alice transfers the CDXJ generated from pushing her Facebook WARCs to IPFS via ipwb (specifying the encryption flag), she may then transfer the CDXJ to Carol. Carol can then decrypt the payload when replaying the mementos described in the CDXJ.

[163]. We evaluated the feasibility of leveraging previous browser-based IPFS implementation through prototypical implementation extending Mink to leverage this sort of communication. 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. We describe this further in Section 8.3.4.

7.3 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 76 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 7.3.1)

Pattern 2: Aggregation of multiple Web archives (Section 7.3.2)

Pattern 3: Aggregator chaining (Section 7.3.3)

Pattern 4: Aggregation with authentication (Section 7.3.4)

Pattern 5: Aggregation including a hybrid public-private archive (Section 7.3.5)

Pattern 6: Aggregation with filtering via MMA interaction (Section 7.3.6)

Pattern 7: Aggregation with filtering via SG interaction (Section 7.3.7)

7.3.1 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 (Chapter 1), we sent a request to a URI-M[2] for the URI-R `nasa.gov`. Figure 77b shows the archived representation, a familiar Web page representation, that is returned to a user when accessing this particular URI-M. Figure 77a 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 4.3) 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.

7.3.2 PATTERN 2: AGGREGATION OF 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

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

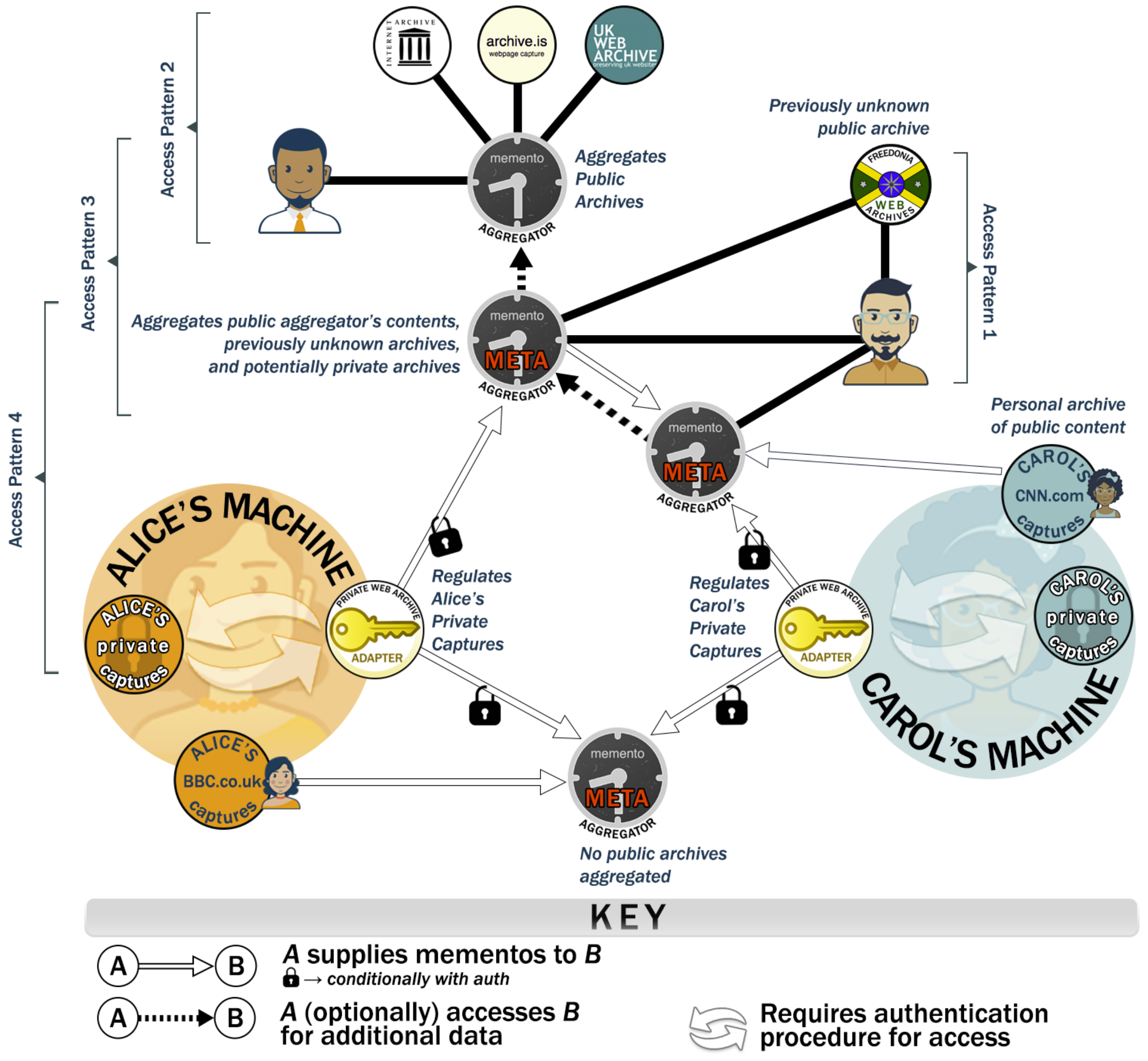

Access Pattern 2
Access Pattern 3
Access Pattern 4
Access Pattern 1
Access Pattern 5
archive.is
UK WEB ARCHIVE
Aggregates Public Archives
Previously unknown public archive
Aggregates public aggregator's contents, previously unknown archives, and potentially private archives
Personal archive of public content
ALICE'S MACHINE
CAROL'S MACHINE
ALICE'S private captures
CAROL'S CNN.com captures
CAROL'S private captures
ALICE'S BBC.co.uk captures
Regulates Alice's Private Captures
Regulates Carol's Private Captures
PRIVATE WEB ARCHIVE ADAPTER
memento META AGGREGATOR
No public archives aggregated
KEY
A supplies mementos to B
→ conditionally with auth
A (optionally) accesses B for additional data
Requires authentication procedure for access

Fig. 76 MMAs and PWAAs form a hierarchy of access for a variety of scopes of Web archives. User Access Patterns from Section 7.3 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.).

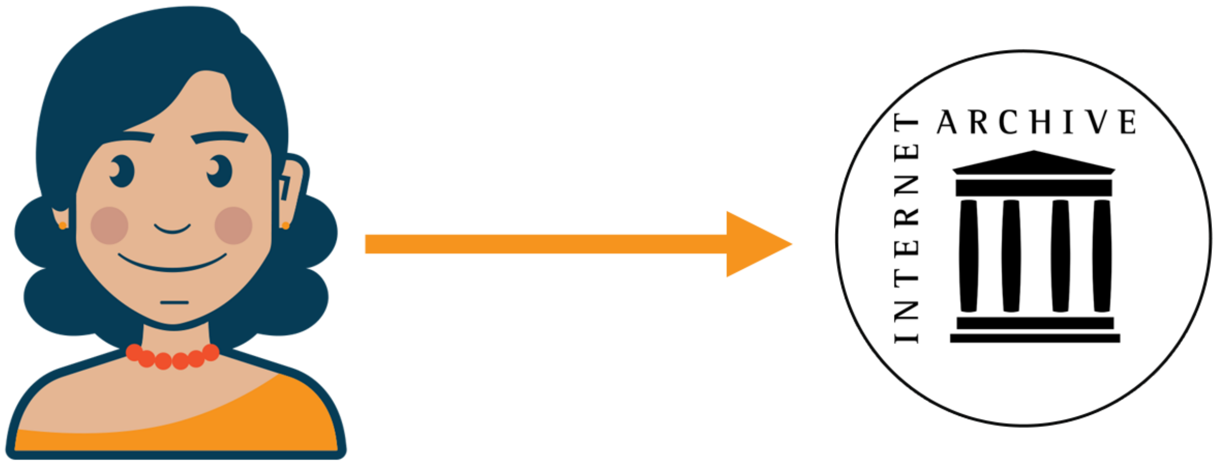
return the TimeMap shown in Figure 24 (Chapter 2). Note the URI-Ms listed are from a variety of public Web archives. Memento proxies [1] also exist to adapt the responses from Web archives that have not yet implemented Memento. Figure 78 shows a user accessing a public Memento aggregator, which aggregates captures from three public Web archives (IA, UKWA, and archive.is). The "1" in this figure indicates that all requests are sent relatively simultaneously with any characteristic of query precedence. Beyond Patterns 1 and 2 resides the contribution of the Mementity Framework.

7.3.3 PATTERN 3: AGGREGATOR CHAINING

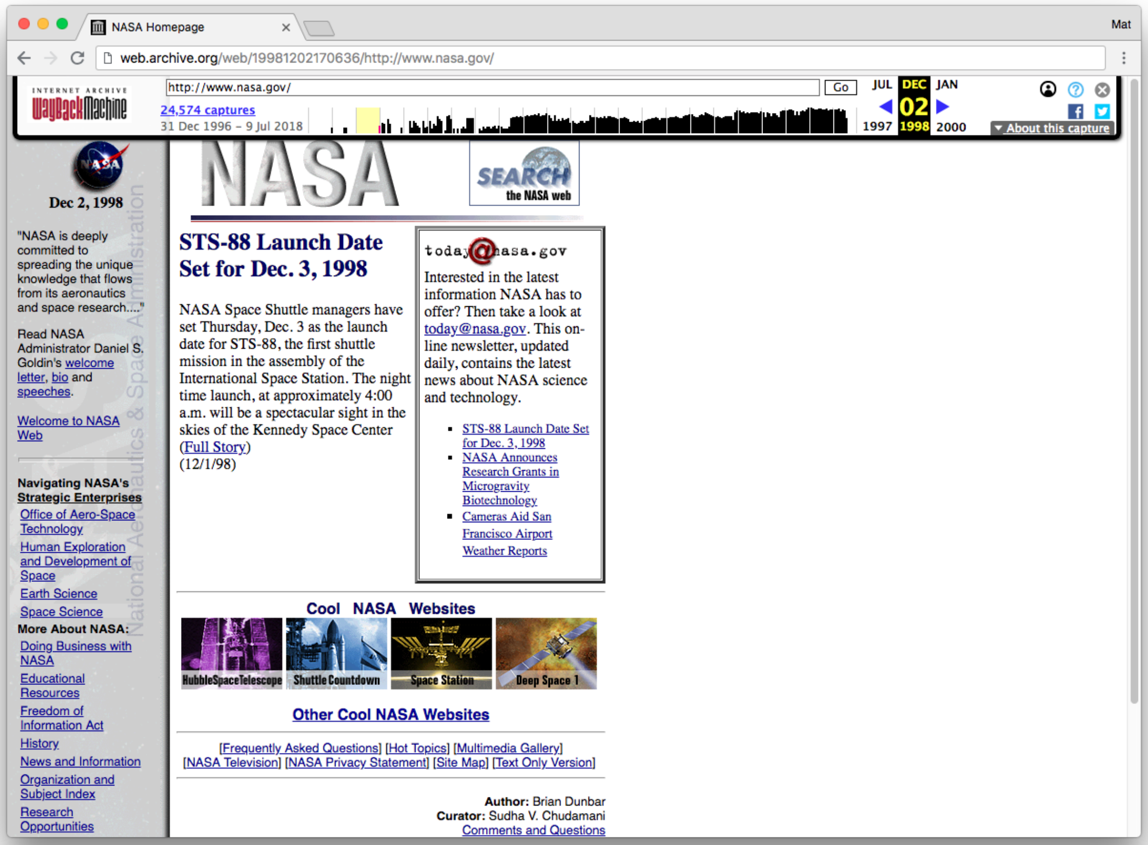
A user may initially access an MMA instead of an MA per Pattern 2. Figure 79 pictorially describes an MMA relaying a request for URI-Ms for a URI-R from a user to the aforementioned MA. The MA performs the query and returns the results to the MMA. The MMA then relays the results to the user. This pattern introduces simple hierarchical chaining of aggregators and is novel to the introduction of an MMA. In the scenario described in Section 7.1.1, 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 7.1.1, a MMA is a functional superset of an MA. Because of this, the MA in Figure 79 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.

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. 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 80 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.



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



Dec 2, 1998

"NASA is deeply committed to spreading the unique knowledge that flows from its aeronautics and space research...."

Read NASA Administrator Daniel S. Goldin's welcome letter, bio and speeches.

Welcome to NASA Web

Navigating NASA's Strategic Enterprises
Office of Aero-Space Technology
Human Exploration and Development of Space
Earth Science
Space Science

More About NASA:
Doing Business with NASA
Educational Resources
Freedom of Information Act
History
News and Information
Organization and Subject Index
Research Opportunities

STS-88 Launch Date Set for Dec. 3, 1998

NASA Space Shuttle managers have set Thursday, Dec. 3 as the launch date for STS-88, the first shuttle mission in the assembly of the International Space Station. The night time launch, at approximately 4:00 a.m. will be a spectacular sight in the skies of the Kennedy Space Center (Full Story)
(12/1/98)

today@nasa.gov

Interested in the latest information NASA has to offer? Then take a look at today@nasa.gov. This on-line newsletter, updated daily, contains the latest news about NASA science and technology.

- STS-88 Launch Date Set for Dec. 3, 1998
- NASA Announces Research Grants in Microgravity Biotechnology
- Cameras Aid San Francisco Airport Weather Reports

Cool NASA Websites

HubbleSpaceTelescope Shuttle Countdown Space Station Deep Space 1

Other Cool NASA Websites

[Frequently Asked Questions] [Hot Topics] [Multimedia Gallery]
[NASA Television] [NASA Privacy Statement] [Site Map] [Text Only Version]

Author: Brian Dunbar
Curator: Sudha V. Chudamani
Comments and Questions

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

Fig. 77 Access Pattern 1 (Section 7.3.1) 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).

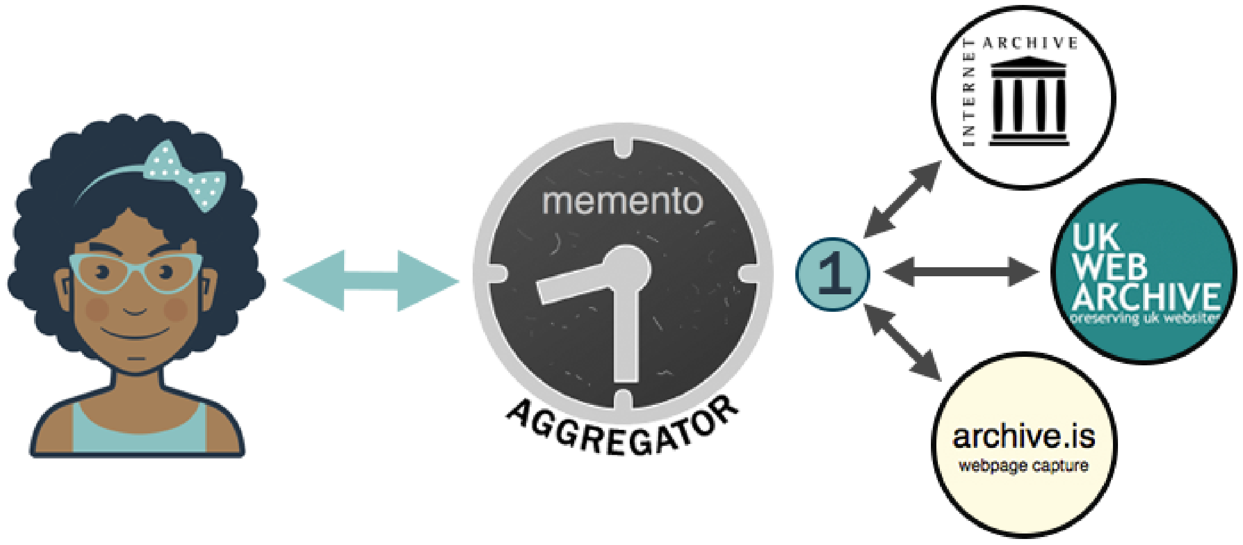


Fig. 78 Access Pattern 2 (Section 7.3.2) represents a user accessing a Memento aggregator to obtain aggregated results from a set of archives. The archives contained in this set are often not customizable by the user. A TimeMap is returned to the user containing URI-Ms and other Memento metadata (e.g., original URI-R). A user may then access a URI-M contained in the returned TimeMap (Section 7.3.1). This pattern exhibits an equally-weighted querying model without precedence (requests are executed in parallel) or short-circuiting.

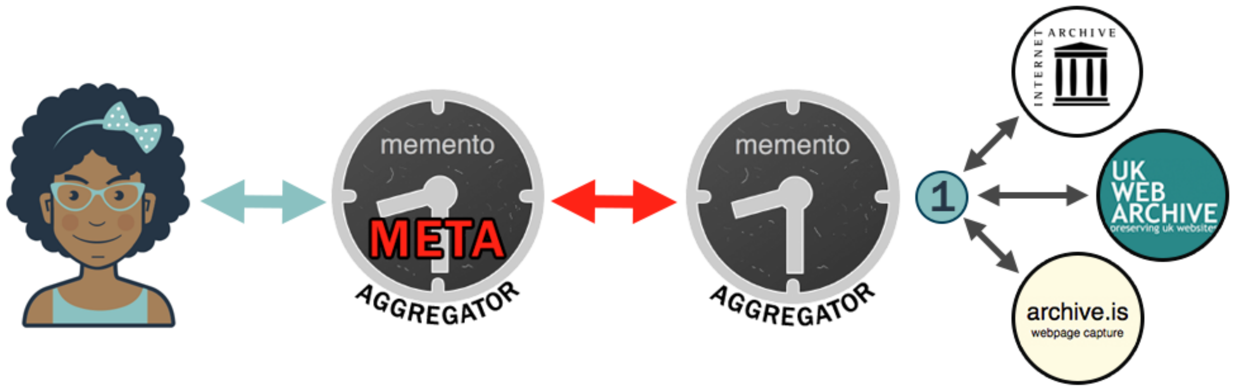


Fig. 79 A Memento Meta-Aggregator (MMA) acts as a functional superset for a conventional Memento Aggregator (MA). This attribute allows an MMA to replace an MA with extended features beyond the scope of a conventional MA. An MMA can also acts as a simple relay of the results (pictured) with the potential 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.

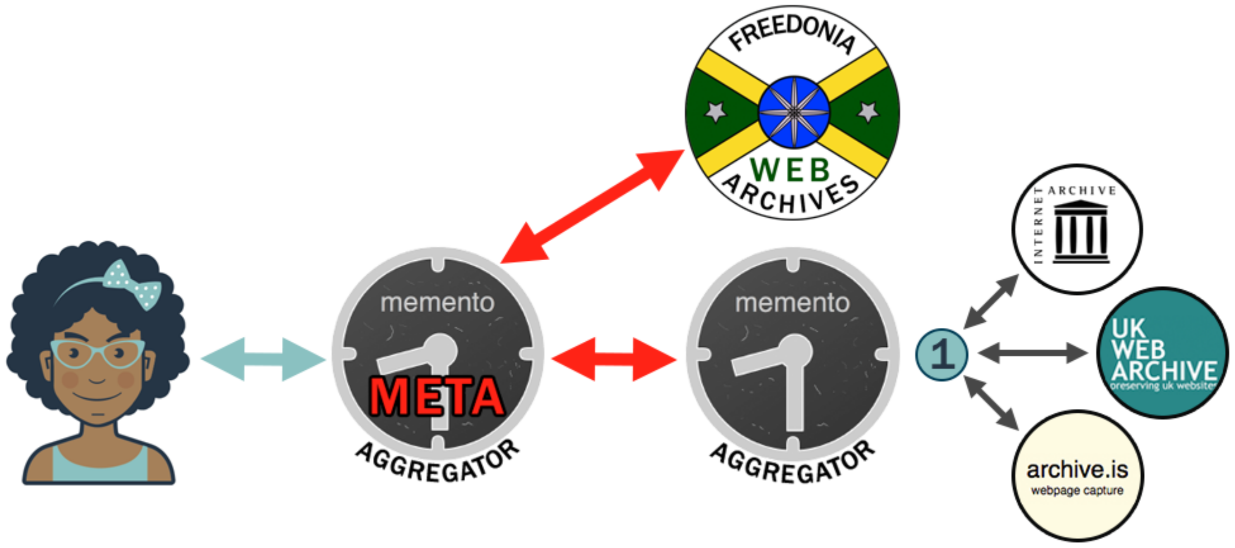


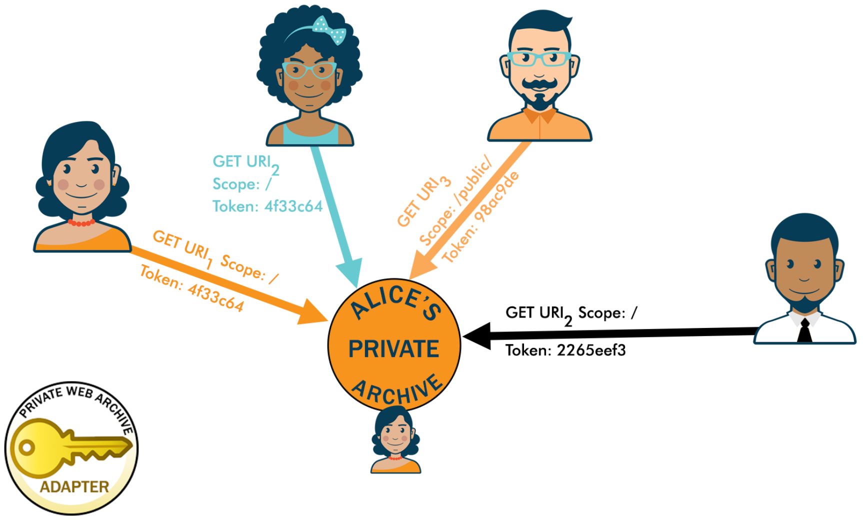
Fig. 80 Chaining Memento Meta-Aggregators allows results to be supplemented. Using a hierarchical MMA approach, a previously unaggregated public Web archive may be aggregated with the results for a URI-R from a conventional Memento aggregator. Pattern 4 extends on this base relationship between MMAs (shown in Figure 79) 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.

7.3.4 PATTERN 4: AGGREGATION WITH AUTHENTICATION

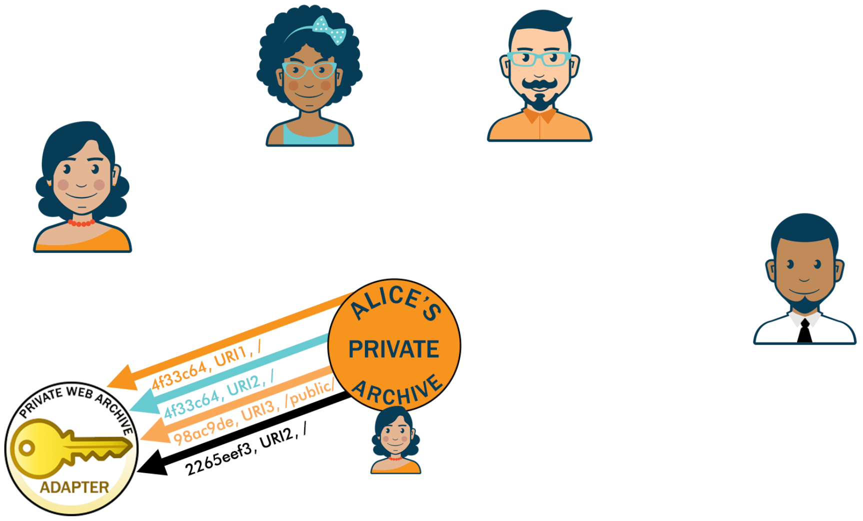
All previous patterns in this section have been described with the assumption that an archive will return a TimeMap of its captures for a URI-R or a URI-M if supplied a URI-R and datetime. As previously discussed (e.g., scenarios in Section 1.2, enabling the personal Web archivist in Section 4), aggregating 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 7.1.2). Because of this, the fundamental Access Pattern 1 does not apply. Bob will experience the authentication flow described in Figure 63 (Figure 6) 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 81). Upon the archive receiving a request from Alice or Bob, who have separately authenticated, or Carol, who is reusing a token (Figure 81), 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 81), who may be rejected access if a token is not supplied or an invalid token is supplied (as configured).

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 82. 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 82a) 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 82b). Both Alice and Carol's PWAA validate their respective keys but Bob's PWAA rejects the key supplied by Alice (Figure 82c). 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 82d). Bob's archive, having

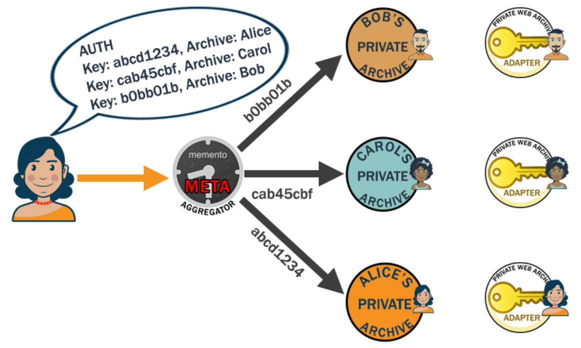


(a) Four users (left to right: Alice, Carol, Bob, Malcolm) request URIs in Alice's private archive using pre-established tokens and a defined "scope". Scope here is reused from the OAuth specification to potentially limit access to parts of an archive on a token basis.

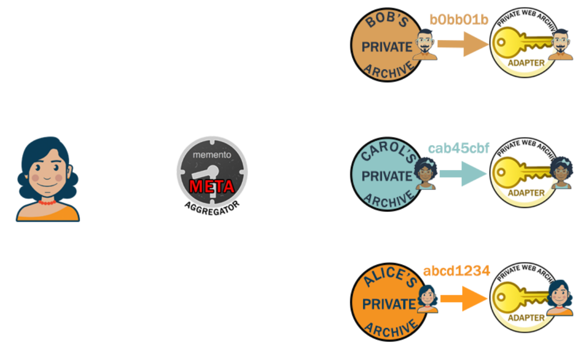


(b) Prior to authorizing access, a private archive will consult its PWAA to verify access to the scope and URI using the respective token suppplied.

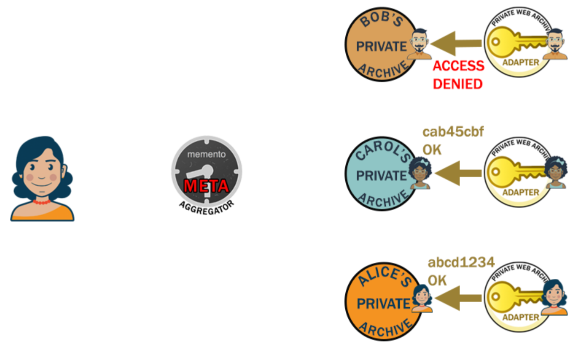
Fig. 81 A token obtained from the process in Figure 63 (Chapter 6) can be shared and reused for persistent access. Accessing the PWAA responsible for access control of a private Web archive will initially deny access without providing credentials. Tokens may be revoked and re-established, allowing regulation of access to private archives. Requests shown as temporally parallel for graphical simplicity but more likely performed at different times.



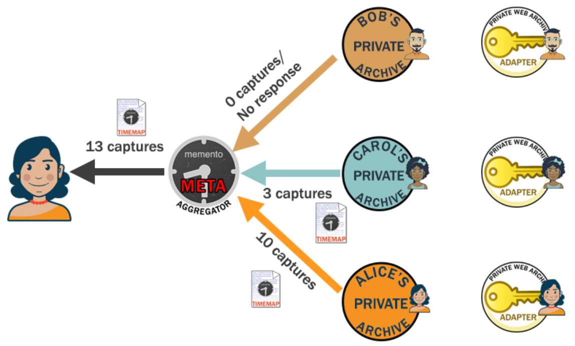
(a) A user provides different keys for three different private Web archives, aggregated by an MMA



(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. 82 In instances where an MMA is configured to only aggregate private Web archives or the *privateOnly* short-circuiting (Section 7.2.2) directive is supplied, a user may specify different keys on a per-archive basis.

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.

7.3.5 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 83). 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 7.1.1) unless URI-Rs are opaque (e.g., `archive.is` obfuscates the URI-R so it cannot be extracted solely from the URI-M). 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.

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

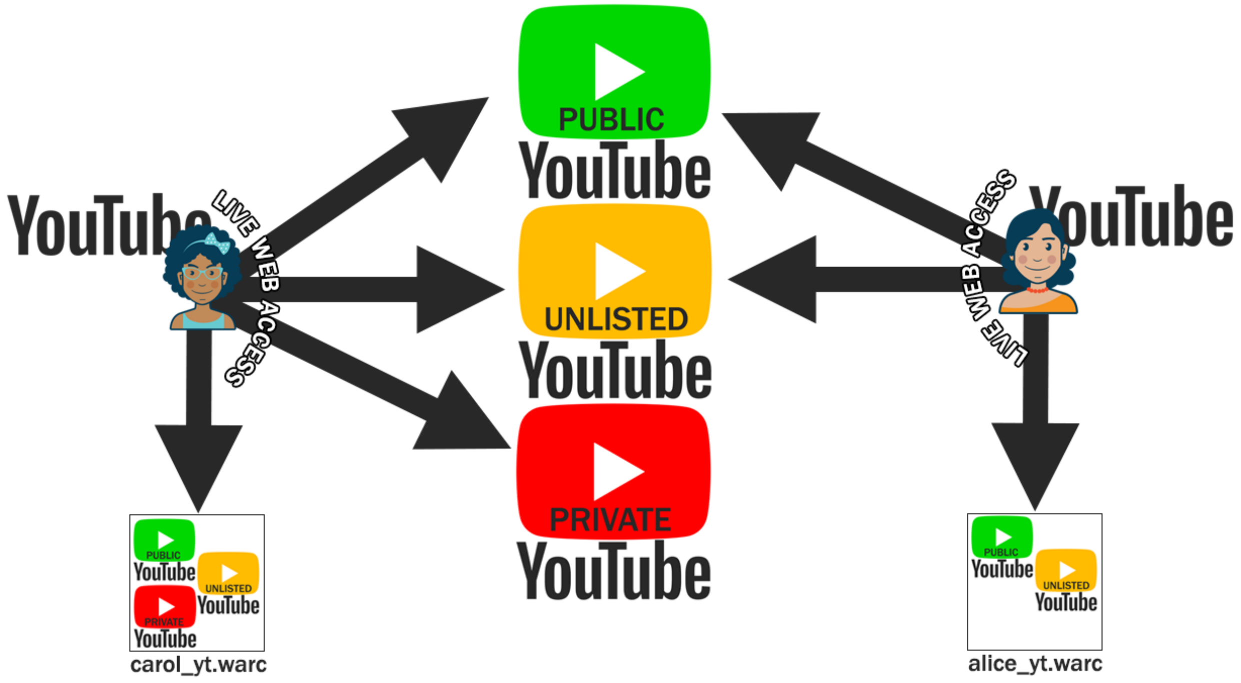
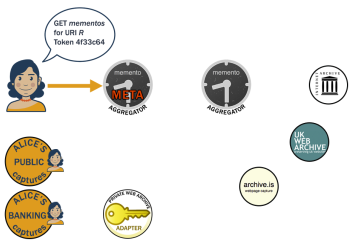


Fig. 83 A user may want finer grained access control of captures within her archive without the need for separate collections. Carol has preserved her public, private, and unlisted `youtube.com` videos but may wish to restrict each class's accessibility within their archive. Alice can access the public and unlisted videos but the question remains as to whether the unlisted video should be publicly available on the archived Web.

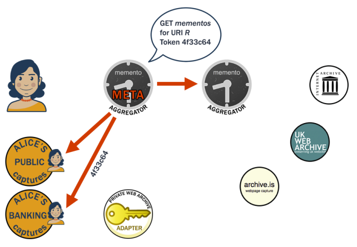
the live Web on a basis of her choice. Alice has setup an MMA to aggregate her archive's captures. She is content with some of the videos being publicly accessible from her archive (e.g., public videos, green icons in Figure 83) and some being selectively accessible to those who authenticate with her archive's PWAA (e.g., yellow and red icons in Figure 83), despite all videos being private on the live Web. Carol, Alice's sister, wishes to utilize Alice's MMA to view some of Alice's private family videos in context 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 70. 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 70 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.

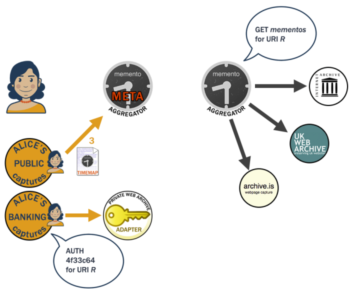
Each archive in the set of archives aggregated by an MMA may require special handling, as exhibited by the aforementioned YouTube scenarios. Figure 84 depicts Alice accessing an MMA that in turn retrieves captures for an MA and Alice's own captures. Alice's captures here, despite being in the same "archive", 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 84a), which is only relayed to the corresponding archive (Figure 84b) and not propagated to where it is inapplicable. While Alice's archive may asynchronously return three results (Figure 84c) 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 63 and 81). Upon successful authentication by the PWAA (Figure 84d), 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 84e 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 84f) and returned to Alice in a StarMap containing 10,143 URI-Ms.



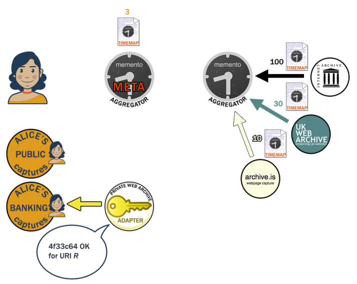
(a) User supplies a URI-R and a pre-obtained token after performing the procedure in Figure 63.



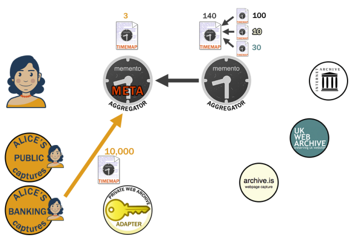
(b) The URI-R and token are relayed (where applicable) from the MMA to the mementities (two archives and an MA).



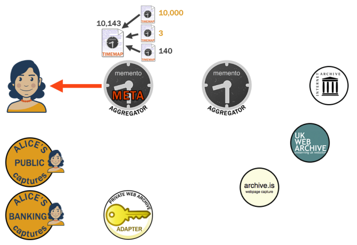
(c) MA requests URI-R from archives, one archive returns results directly to MMA, a private archive verifies the token with an associated PWAA.



(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. 84 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.

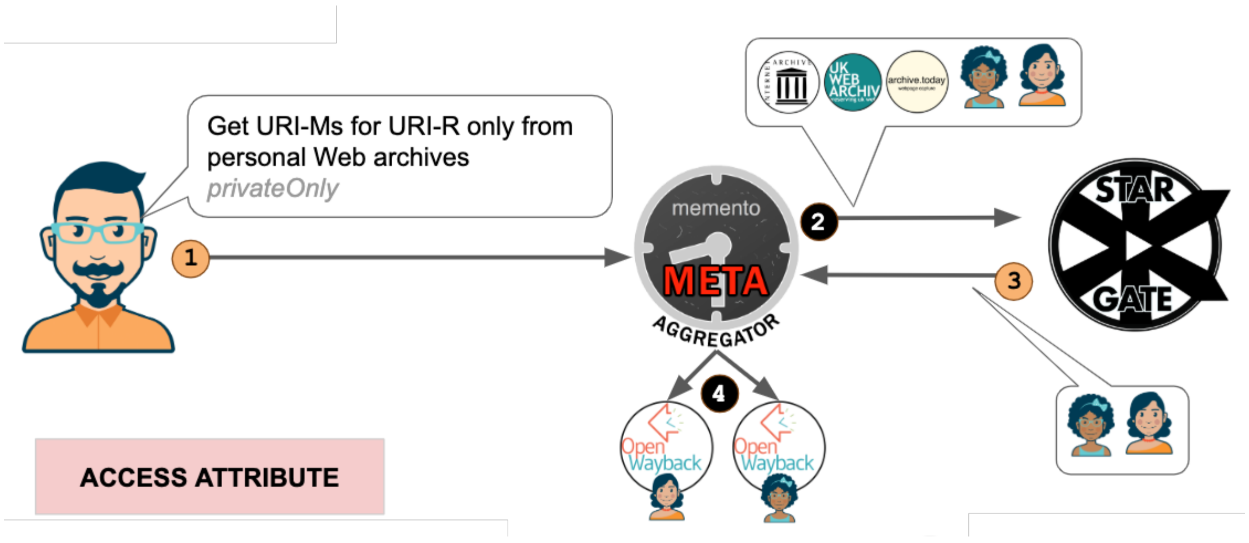


Fig. 85 Bob's request captures for a URI-R to only be requested from archives that meet the "private" access attribute (marker 1). The MMA then relays this request with its configured archives to a SG (marker 2), which the SG filters and sends back as a list to the MMA (marker 3), which the MMA then queries (marker 4).

7.3.6 PATTERN 6: AGGREGATION WITH FILTERING VIA MMA INTERACTION

In Section 7.2.1 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 6.2.

Pattern 6a involves pre-filtering URI-Ms based on access attributes. In this sub-pattern, Bob sends a request for a URI-R to an MMA with the HTTP request header `Prefer: privateOnly` (Figure 85). The MMA send a request to an SG with the archives it supports for aggregation $\{A_0\}$ and relays the `Prefer: privateOnly` (marker 2) supplied by Bob (marker 1). 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 (marker 3). On the basis of this refined set of archives, the MMA can then perform the procedure described in Pattern 4, Figure 82, and marker 4 where each respective private Web archive may require an authentication procedure and a respective PWAA. This exhibits negotiation in a dimension beyond time, i.e., on the access attributes as described in Section 6.2.3.

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, Bob

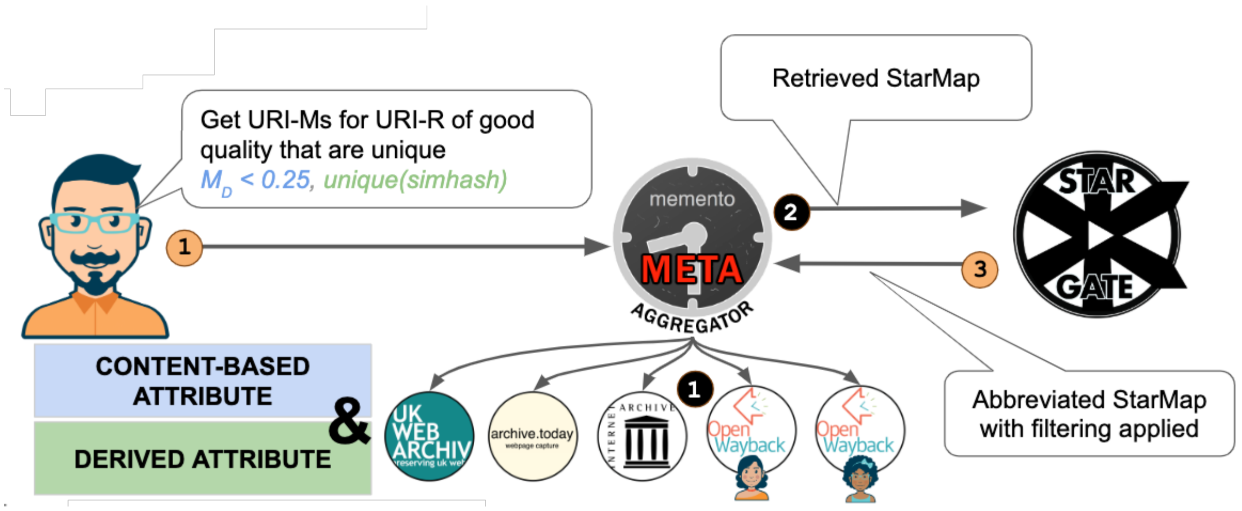


Fig. 86 Bob's request captures for a URI-R that have Memento Damage below a threshold and are unique based on their SimHashes (marker 1). The MMA initially obtains the URI-Ms from the archives and polls the relevant services (not shown: a damage and simhash calculation Web services). This StarMap is then passed to a StarGate with the criteria for filtering (marker 2) then subsequently filtered by the SG and returned to the MMA (marker 3). The filtered StarMap is returned to Bob.

sends a request to an MMA with an HTTP request header of `Prefer: damage"<0.25"` and `Prefer: unique(simhash)`. An MMA may recognize damage as a derived attribute and first send a request for a TimeMap for the URI-R Bob 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"<0.25"` 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"<0.25"`. 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 62. This StarMap is then returned to the MMA and relayed to Bob.

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 describe further in Section 8.2. 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 StarMap is returned to the MMA and from there returned to Alice.

7.3.7 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"<0.25"` just as Bob 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 6.1. 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 StarMap, 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 StarMap only containing the mementos' identifiers and respective attributes that met the condition Bob specified, prepends the StarMap with the metadata information as in Pattern 6b, and returns the StarMap to Bob.

The set of access patterns can be summarized as follows.

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

7.4 FRAMEWORK EXTENSIBILITY

The mementities in Section 7.1 are designed to be applicable to a variety of existing user access patterns (some described in Section 7.3) with the intention of further applicability beyond the extent explored in this dissertation. To ensure this, we have designed the mementities in this dissertation 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 7.3.4 and Figure 84), conventional Memento Aggregators (Section 7.3.3 and Figure 79), 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 7.1 performed a single role in the hierarchical relation of each other respective mementity. In Section 7.3 we described seven access patterns, with the final five leveraging the new capability of the mementities for aggregating private and public Web archives. It might be the case that beyond the initial framework defined in this dissertation, other roles are necessary to account for the dynamics of some Web archives. 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 dissertation to remain functionally cohesive.

7.5 SUMMARY

In this chapter we defined the core mementities and fundamental dynamics of a Framework for aggregating private and public Web archives. Section 7.1 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 7.2 described specific dynamics of the mementities and how they interact to form the hierarchical behavior of the framework. In Section 7.3 we built upon conventional access patterns to integrate usage of the mementities in Section 7.1, tying in the usage with real-world scenarios. In Section 7.4 we discussed the extensibility of the Mementity Framework to ensure that it is adaptable to unforeseen scenarios and dynamics in Web archiving.

CHAPTER 8

FRAMEWORK EVALUATION

It is only when we have renounced our preoccupation with "I," "me," "mine," that we can truly possess the world in which we live. Everything, provided that we regard nothing as property. And not only is everything ours; it is also everybody else's.

- Aldous Huxley, *The Perennial Philosophy [93]*

Evaluation of the Mementity Framework described in this dissertation is multi-fold. In Section 8.1 we evaluate the design decisions for each of the mementities. In Section 8.2 we investigate costs for enrichment of TimeMaps to generate StarMaps. In Section 8.3 we describe our reference implementation for the Mementity Framework. Finally, in Section 8.4 we evaluate how the scenarios described in Chapter 1 can be realized and resolved using the Mementity Framework introduced in this dissertation.

8.1 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. Because the Framework was progressively developed, 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.

As a review, RQ1 and RQ2 deal with the preservation and replay process independent of Memento. Our studies in *archivability* (Chapter 5) identified content that was difficult to capture and replay. Our tools to capture content behind authentication (RQ3, Chapter 4) mitigated some of these issues, leveraging Web browser APIs to increase archival quality (RQ2). Preserving this private content set the basis for RQ4, RQ5, and RQ6.

In Chapter 6 we focused on the integration of private and public Web archives with a focus on how content that was preserved behind authentication (e.g., with the tools in Chapter 4) could be systematically aggregated. The addition of the StarMap concept and the attribution of privacy-related attributes (Section 6.2.3) to relevant URI-Ms provides a means of resolving RQ4. The PWAA mementity (Section 7.1.2) provides an integrative means of executing this special handling (RQ6) with the introduction of a more capable Memento aggregator (MMAs, Section 7.1.1) allowing for this access flow to be performed from an aggregate perspective (RQ5). With the addition of richer definitions for mementos, advanced content negotiation for privacy and an arbitrary, extensible set of other attributes is facilitated by the StarGate (Section 7.1.3) for potential use cases beyond this dissertation.

8.2 COSTS OF GENERATING STARMAPS (AND LINK)

In Section 6.2 we discussed how adding additional attributes to memento descriptions 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 include 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.

With requests to a conventional Memento aggregator for a URI-R, an estimated temporal cost (T, Equation 9) can be calculated by considering the time to send the request to the aggregator t_{req}, the aggregator communicating with each archive (t_{MA}), the aggregator aggregating the responses (t_{AGG}), and the response being returned to the client (t_{resp}).

$$T = t_{req} + t_{MA} + t_{AGG} + t_{resp} \tag{9}$$

.

The time required for an aggregator to request the TimeMap from an archive (A_i) may vary based on the communication speed of the slowest responding archive as well as potentially being inversely proportional to the number of holdings. For example, an archive may take longer to assemble the relevant URI-Ms from its CDX index (Section 2.4.4). This roundtrip time (RTT) estimate assumes a query model without precedence (Section 7.2.2) for the order of request, is exhibited by all Memento aggregators by default, and is discussed more in Section 8.3.1. Equation 10 describes the time required to obtain all TimeMaps from the configured archives from an aggregator's perspective.

$$t_{MA} = max(RTT(A_i, \mathit{URI\text{-}R})) \tag{10}$$

The aggregation algorithm implemented by an aggregator affects the temporal complexity of t_{AGG}. MemGator sorts archives' individual TimeMaps as they are received[1] and progressively aggregates and sorts them with an optimization for using smaller TimeMaps for comparison to account for sorting efficiency. This optimization is based on a temporally sparse archive with fewer URI-Ms being exhausted in the iteration procedure quicker than a large TimeMap, i.e., for combining two TimeMaps, only one of the two needs to be traversed for sorting before the remainder is appended. For example, archive A_1 has m mementos for a URI-R from the years 1996 to 2000. A_2 has n mementos for a URI-R from the years 1999 to 2019. Once a TimeMap is fetched, if n is much greater than m in the number of URI-Ms in its TimeMap, m would be used as the basis of traversal. In this scenario, the pivot indexing the URI-Ms in A_2 in the sorting procedure would only advance partially through the list before the remainder of the list is simply appended, due to the nature of temporal sorting. In the worst case scenario, $m + n$ iterations will occur but in cases of sparse archives, having the archive with fewer captures of a URI-R as the basis of iteration will invoke this appending procedure and make the sorting procedure resolve more quickly.

The additional capabilities of an MMA beyond MemGator requires the parsing and interpretation of client side preference (t_{prefer}), generation of attributes where applicable (via communicating with TimeMap enrichment Web services, t_{enrich}), and filtering the results as requested using HTTP `Prefer` (t_{filter}). Parsing and filtering are both handled by the StarGate, so should not incur spatial costs beyond caching and ought to be computationally straightforward with linear complexity. The temporal costs associated with t_{enrich} will be dependent on the services and incur additional communication costs like t_{req} and t_{resp} when communicating between the StarGate and the relevant service.

As an example, a client can request only archives being considered that are `privateOnly` using `Prefer` (t_{req}). A StarGate receives this request, and sends its preconfigured set of archives (given that the client did not specify a custom list in this example, which would otherwise replace the basis set) along with the parameters for the `privateOnly` preference to a service to interpret the parameters (details in Section 8.3.3) and respond to the

[1]TimeMaps are conventionally sorted by time but this is not mandated by Memento.

StarGate, with transmission time bundled into t_{enrich}. In this scenario, no further filtering is needed by the StarGate and the filtered response can be returned. If the Preference was instead specified to a service that calculated a value for a derived attribute for a URI-M (e.g., Memento Damage), the StarGate may acquire all values for the URI-R (i.e., for each URI-M) then perform the filtering procedure, as specified by the client's preference. This pre- and post-filtering procedure is described further in Section 7.3.6.

8.3 EVALUATION THROUGH IMPLEMENTATION

In one effort to evaluate the framework, we implemented the mementities in Section 7.1 by both extending existing software (MemGator and Mink, Section 8.3.1 and 8.3.4, respectively) and creating new software packages (PWAA and StarGate, Sections 8.3.2 and 8.3.3, respectively). Of note here is our adaptation of software to utilize the reference mementity implementations of the framework, i.e., Mink, 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 7.2.3). This approach provided 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, provided a more realistic use case of the Mementity Framework. This also facilitated further evaluation of the user-experience of the framework and allowed us to more comprehensively answer RQ2.

To facilitate archive collaboration (Section 4.5), we also extended Mink beyond simply interfacing with the mementity implementations by encouraging collaboration of personal and private captures through sharing and interfacing of personal Web archives (Section 7.2.4). We used 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 utilized the work done with the JavaScript implementation of IPFS [163].

The remainder of this section describes the implementations of the various tools to exhibit the Mementity Framework. The reference implementations for the three mementities described in Section 7.1 (Memento Meta-Aggregator, Private Web Archive Adapter, and StarGate) are detailed in Sections 8.3.1, 8.3.2, and 8.3.3, respectively. The MMA basis, MemGator, was initially programmed in the Go programming language [66]. Go currently offers a wide level of support for generating cross-platform binaries (mitigating end-user compatibility issues), so we followed suit and programmed the mementity reference implementations in Go as well. Section 8.3.4 describes advancements in capability of the Mink

	Nomenclature	Interpretation
N_1	$[\{A_0\},\{A_1\},\{A_2\}]$	Sequence of three archives to be queried in-order. MemGator interprets this like N_2.
N_2	$\{\{A_0\},\{A_1\},\{A_2\}\}$	Three archives to be queried simultaneously without precedence.
N_3	$[\{A_0\},\{A_4,A_2,A_5\},\{A_3\}]$	Three sets of archives to be queried in series while also pseudo-parallel within a set.
N_4	$\{\texttt{"id0"}:\{A_0\},\texttt{"id1"}:\{A_1\},\texttt{"id2"}:\{A_2\}\}$	Added identifiers for archive specifications in N_2
N_5	$[\{\texttt{"id0"}:\{A_0\}\},\{\texttt{"id1"}:\{A_1\}\},\{\texttt{"id2"}:\{A_2\}\}]$	Seemingly verbose identifiers for single-archive sets.

Table 6 The evolution of archival sets and implied precedence can be interpreted from the syntax and semantics of the JSON definition.

browser extension with respect to enhancing the degree of client-side interaction with Web archives (Section 6.1).

8.3.1 MEMENTO META-AGGREGATOR

MemGator serves as the basis for the MMA reference implementation. A nuance encountered in the original MemGator design stems from the interpretation of the JSON configuration used in aggregation in that it does not interpret an "array" of archive specifications (Nomenclature N_1) as having query order. Table 6 describes a progression of "Nomenclatures" abstracting archival specifications and how they should be interpreted per the JSON specification. As with most Memento aggregators, MemGator queries all archives simultaneously (i.e., in "series" as allowed with HTTP without regard to order) with the exception of those that have been disabled within the configuration or have been explicitly set to be ignored in the consumed archival specification. Here we want to emphasize that despite the configuration (archival specification used by MemGator) resembling Nomenclature N_1 (where A_n is representative of configuration attributes for an archive like in Appendix B), the functionality exhibited in MemGator is more in line with Nomenclature N_2. Our extension of MemGator modifies the specification interpretation by MemGator to align with the JSON specification [45] in that JSON arrays imply order.

Request order is significant in the Mementity Framework, as providing this ability allows for query precedence and short-circuiting (Section 7.2.2). A specification syntactically structured like Nomenclature N_1 would cause A_0 to be queried, then A_1, then A_2. With a specification exhibiting Nomenclature N_2, all three archives would be simultaneously queried without expectation of order (as occurs in MemGator currently). Exhibition of Nomenclature N_3 from an aggregator that correctly interprets the JSON syntax (i.e., not MemGator prior to this dissertation) would first query A_0 then, once complete, simultaneously query A_4, A_2, and A_5 and only then query A_3. As an aside, though contemporary JavaScript has the idiom of Sets [70], it has not yet been adapted to JSON.

To exhibit the capabilities of an MMA as described in Section 7.1.1, we extended MemGator to meet the following MMA deliverables (MMAD):

MMAD$_1$: Allow for the requests to be sent to sets of archives ($n \geq 1$) to be performed sequentially ("query precedence" per Section 7.2.2)

MMAD$_2$: Allow for ceasing further requests from being sent in a sequence of archival sets if a condition is met ("short-circuiting" also per Section 7.2.2)

MMAD$_3$: Consume a set of archives specified by a client as the basis for aggregation ("client-side archival specification" per Section 6.1)

MMAD$_4$: Communicate with PWAAs for authentication and access by relaying tokens to private archives for aggregation ("aggregation with authentication" per Section 7.3.4)

MMAD$_5$: Respond to negotiation in other dimensions through communication with a StarGate ("aggregation with filtering via MMA interaction" per Section 7.3.6)

Expressing Order in an Archival Specification

MemGator's current archival query algorithm (Figure 87) uses a static set of archives as the basis for aggregation. The algorithm *does* consider an absolute archival count for short-circuiting, but this is server-configured (by whomever is running the instance), static (cannot be changed without restarting the instance), and based solely on archival "order". MemGator reads the configuration file and sorts the archives using a server-specified "probability" float value, of which the basis is undisclosed in the code. The aforementioned sort order, as defined by this probability within the configuration file, is static to the server instance and not customizable by the client.

Following sorting, MemGator then skips `dormant` archives (as expressed by an associated boolean value in the archival specification) then adds all archives to a list to maintain synchronicity to ensure all archives have responded, timed out, or the process to fetch the archive's TimeMap has otherwise resolved. The respective archive is then queried based

```
int absoluteArchiveCount_shortCircuit
array timemaps[]
array archives[n]

foreach i, archive in archives:
  if i == absoluteArchiveCount_shortCircuit
    breakLoop
  if isDormant(archive):
    continue
  timemaps[] += async fetchTimeMap(archive.TimeMapEndpoint, URI-R)

aggregatedTM = []
async when responseReceived(archive):
  baseTM = archive.timemap
  if sizeOf(baseTM) < aggregatedTM:
    aggregatedTM, baseTM = baseTM, aggregatedTM
  int pivotBase = baseTM.firstMemento
  int pivotAggr = aggregatedTM.firstMemento
  for mementos in baseTM:
    if baseTM[pivotBase].datetime > aggregatedTM[pivotAggr].datetime:
      aggregatedTM.insert(baseTM[privotBase])
      pivotBase = baseTM.next
    else:
      pivotAggr = aggregatedTM.next
defer return aggregatedTM
```

Fig. 87 This pseudocode is a simplified representation of MemGator's current TimeMap aggregation algorithm. Fetching TimeMaps is pseudo-parallel while an aggregated TimeMap is progressively built by the aggregator as Web archives respond to the respective request.

```
aggregateTimeMaps(session):  // archives to aggregate, defined in "session"
  (Figure 87)

aggregateSetsOfTimeMaps(urir, session):
  Archives[] wholeSet = session.archives
  TimeMap aggregatedTimeMap
  foreach archivalSet in session.archives:
    session.archives = archivalSet
    aggregatedTimeMap += aggregate(session)
  aggregatedTimeMap.sort()
  session.archives = wholeSet
  return aggregatedTimeMap
```

Fig. 88 A method to reuse the aggregation algorithm in Figure 87, but allow for precedence, is to subset the entire set of archives and supply the set as a basis to aggregate. This will regress into a single set of archives (and thus, all archives being aggregated per prior MemGator functionality) if the JSON member notation is used instead of JSON arrays (which imply precedence).

on the order within the list (descending "probability") but done so as quickly as the loop can iterate, i.e., a subsequent archive is not required to wait until a response from the previously queried archive has been received. Despite this programmatic order of querying, the requests are essentially executed in parallel (or *pseudo-parallel*). This is evident in the implementation waiting until a decrementing counter for archives has resolved, thus treating the set of requests as independent without order. Our extended implementation of querying TimeMaps (Figure 88) allows for the current behavior of pseudo-parallel requests (as expected for multi-query efficiency) as well as exhibiting requests performed sequentially or in "explicit series" where the previous requests must resolve prior to proceeding ($MMAD_1$). Though this inherently increases the amount of time required for all requests to resolve, explicit series requests allow for short-circuiting ($MMAD_2$) and are the ethos of the query precedence concept introduced in the Mementity Framework.

Modifying MemGator to Receive Client-Side Archival Specification

MMAD$_3$ required a significant adaptation of MemGator similar to the preliminary `X-Archives` implementation in Section 6.1.1 but more systematic as described in Section 6.1.2 (using `Prefer`). As above, MemGator uses a static set of archives that is programmatically global to the code (and thus, MemGator instance) instead of associating the set with the session, which is representative of the request by a client. Despite this, the initial MemGator implementation contained a `Session` data structure to which it only associated time of request to have a temporally-based, sufficiently granular identifier for communication with the client.

The initial adaptation required first associating the default list of archives, as read from MemGator's archival specification, with the `Session` instance that is associated with the request received. This per-Session set of archives prevented the basis/global set from being modified for subsequent requests by default, i.e., if a subsequent client does not specify a custom set of archives, the basis set will still be used. Figure 89 shows the high-level modifications required to be made to the algorithm in Figure 87 to allow the aggregator to meet the requirement of MMAD$_3$.

MMAD$_1$ requires revising the algorithm used to consider order of archives queried. With a client's preferences now being considered in Memgator (Figure 89), we made the change to interpret what is currently supplied in MemGator both within the implementation and a sample specification (Nomenclature N$_1$) to the exhibited functionality like Nomenclature N$_2$ to align with the JSON standard[2]. We adapted the archival specification supplied with MemGator to be representative of a JSON object, i.e., changing the JSON from resembling Nomenclature N$_1$ to resembling Nomenclature N$_2$. We then modified MemGator's specification parsing algorithm to expect a JSON object at the broadest scope and associated each archive in the specification with a key, as an object-of-objects is disallowed per the JSON Augmented Backus-Naur Form [59] (ABNF, which defines legal grammar syntax) without a key member (e.g., `"ia"` for the Internet Archive specification of endpoints). This identifier is arbitrary and could also be a key more inline with the hostname, but the semantics of the key should be not interpreted, as a hostname of an archive may change over time. For example, the archive that originally resided at `europarchive.org`[3] moved to the domain `internetmemory.org`[4]. Other archives like `webarchive.proni.go.uk` moved

[2]i.e., arrays imply order

[3]The domain is currently used to promote irrelevant spam content [151].

[4]Following the move to the new domain, this archive has since become inaccessible.

```
type Session struct {
        Start time.Time
        Archives []Archive
}

function aggregateTimeMaps(urir, *session):
  (Figure 88)

function router(httpResponse, httpRequest):
  Archives archives = readFromLocalSpec()
  Session session = (now(), archives)

  string rawPreferHeader = httpRequest.headers["Prefer"]
  strings format, charset, encoding, data = parsePrefer(rawPreferHeader)
  json archiveSpecification = decodePrefer(format, charset, encoding, data)

  if validArchivesSpec(archiveSpecification):
    session.archives = archiveSpecification
  ...
  aggregateSetsOfTimeMaps(httpRequest.urir, session)
```

Fig. 89 MemGator's `router` function handles HTTP requests and responses. This pseudo-code represents changes we made to MemGator to read the HTTP `Prefer` header in a request and use its contents as the basis for which archives to query. To meet the requirement of MMAD$_3$, the aggregator must first consider the set of archives specified by the client as the basis for the set queried.

to using external institutional services for their archival hosting, in this case, Archive-It (Appendix A). Still more, archiving services that have their existence threatened (e.g., their registrar disapproves of their holdings) may be propagated to other top-level domains, as has occurred with the archive that has resided at `archive.today`, `archive.us`, and `archive.md`, among other domains [159, 61]. With this in mind, and to progress the discussion beyond a JSON key member for an archive, we further emphasize that the key is arbitrary and its semantics up to the preference of the user. The revised default specification for MemGator would abstractly resemble Nomenclature N_5, which we reiterate, implies a simultaneous querying model without precedence.

To extend on the revised querying model of MemGator in Nomenclature N_4, we can reinterpret Nomenclature N_1 as having query precedence, per the JSON standard, in that the three archives are represented in an array. This may be further extended to have referenceable identifiers like Nomenclature N_5, which initially seems syntactically verbose. However, with a specification that represents both order (and thus query precedence) of archives to query as well as identifiers for each archival specification, we can begin to add additional archives in the "sets" to be queried (Figure 90) to resemble a more expressive, semantic, and identifiable set of archives to query in series. Figure 90 is similar to the precedence specification of Figure 73 in Chapter 7 (without the client-specified access attribute precedence) where the set of archives consisting of Carol's, Alice's, and Freedonian Archives are first queried (the latter figure used all private archives), then Internet Archive (key `ia`), and finally UK Web Archive. Note the difference in Figure 90 having three sets to its query precedence (Web archives in series) where Figure 73 groups the remaining two public archives to be queried in pseudo-parallel following requests to a set of private Web archives.

Correlative Client-Side Specification Construction

The above modifications to MemGator use the existing model of a static set of archives with tweaks to the semantics, syntax, and interpretation of the archival specification. As described in Section 6.1.2, we implemented the ability for a *client* to specify this sort of archival specification and have MemGator interpret it as intended. As with the examples in Section 6.1.2, this specification can be supplied by a client encoding the JSON and prepending it with necessary descriptors (e.g., `data:application/json;charset=utf-8;base64;`) to ensure its correct, unambiguous interpretation by MemGator. A user (or tool acting as an agent to the user) would base64 encode the JSON in Figure 90, having first defined the endpoints (e.g., URI-Ts) for each archive, into a string like

```
[
  {
    "carol": {
      "timemap": "http://carol.org/tm/"
    },
    "alice": {
      "timemap": "http://archive.alice.org/tm/"
    },
    "freedonia": {
      "timemap": "https://archive.fr/timemap/"
    }
  },
  {
    "ia": {
      "timemap": "http://web.archive.org/web/timemap/link/"
    }
  },
  {
    "ukwa": {
      "timemap":
      ↪ "https://www.webarchive.org.uk/wayback/en/archive/timemap/link/"
    }
  }
]
```

Fig. 90 Sample archival specification with implied query precedence (using a JSON array) and 3 sets of archives (with each set have 3, 1, and 1 item, respectively) to query in-series.

"WwogIHsKICAgICJjYXJvbCI...IKICAgIH0KICB9Cl0=". The client then prepends this string with the aforementioned descriptor, given it is a data URI with a MIME type of application/json using a utf-8 character set that was base64 encoded. Each component of the descriptor can also be modified as the algorithm for encoding and formatting is changed (e.g., the yaml/base58 example in Section 6.1.2). A client would supply this value as the HTTP Prefer header when making requests to an "enhanced MemGator". Upon receipt, MemGator would decode the JSON, parse, and interpret the archival specification as described by the client. This exhibits the dynamics needed for MMAD_3. We implemented this as described above by reading the Prefer request header supplied by a client, overriding the default list of archives if specified, and associating it with the Session instance in MemGator that is passed through the relevant functions (Figure 89).

Short-Circuiting

With MemGator's additional capability of query precedence, a fundamental feature of the MMA mementity is to be able to short-circuit when some condition is met. In Section 8.3.3 we discuss acquiring and associating additional attributes to URI-Ms, which may be the basis for the short-circuiting condition. For example, if a user requests a "memento count" threshold with the counting basis being HTTP 200s (as extensively discussed in our previous work [116]) from a series of archives, archives later in the series may not be queried if the threshold is met. In Section 8.2 we described MemGator's functionality of progressively assembling an aggregated TimeMap as archives respond. In this case, despite archives potentially being queried in pseudo-parallel, the parameter specified by the client may allow late responses to be discarded and the results returned to the client quicker.

MemGator needed to be adapted to understand the semantics for the conditions of short-circuiting, which may be driven by use cases beyond this dissertation. In the above example, the condition of "memento count" as well as "status code" needs be express-able by the client and understood by the aggregator. Figure 71 (Chapter 7) shows an example of a client specifying Preference of a derived attributed (damage) using Prefer. In the example in Figure 71, only URI-Ms would be returned that meet this condition, however, there is nothing specific about this that indicates that the querying should halt before the procedure is exhausted (i.e., short-circuiting). Where damage acts as a per-memento filter, memento-count is descriptive of a threshold condition on the data set.

We modified MemGator to be receptive to requests for this short-circuiting using Prefer. The correlative response header of Preference-Applied allows the implementation to

```
(Figure 88)

  foreach archivalSet in session.archives:
    session.archives = archivalSet
    aggregatedTimeMap += aggregate(session)
    checkIfPreferenceMet(aggregatedTimeMap, preference) ? break : continue

(Figure 88)
```

Fig. 91 This pseudocode replaces a portion of the code in Figure 88 to check whether a condition is met with each iteration of an archival set as specified (e.g., `memento-count`), and if so, stops the querying process through breaking the iteration of archival sets.

indicate whether the short-circuiting was exhibited or if the procedure was exhausted, e.g., whether m mementos for a URI-R from a set of archives could be found and returned. Figure 91 shows an example of the above short-circuiting request specifying `memento-count` above a threshold for mementos with an HTTP `statusCode` of 200.

8.3.2 PRIVATE WEB ARCHIVE ADAPTER

In our reference implementation[5], we created software to exhibit the OAuth 2 protocol for communication and access control for individual private Web archives and when aggregated (Section 7.3.4). In Section 7.3.4 we described an access pattern where users directly access a private Web archive and are directed to an authentication procedure (Section 7.1.2). In this same section we described authentication and sharing of tokens with respect to aggregation (MMAD_4). In this section, we describe the implementation to exhibit these two access patterns.

It is beyond the expected responsibility for a private Web archive to retain credentials for authentication but may it may be expected to look to a second party to validate persistent tokens. We have provided a PWAA Reference implementation in Appendix E. Our reference implementation provides an endpoint for credential registration, which can then be passed to a second endpoint to acquire a token. A private archive can query an additional endpoint to validate the token by interpreting the JSON response supplied by the PWAA. An MMA

[5]`https://github.com/machawk1/pwaa`

can act in the same manner to perform this procedure. This implementation satisfies the requirement of $MMAD_4$.

Figures 81 and 82 (Chapter 7) provide sample uses cases where a user or set of users access a private archive directly through the PWAA or via aggregation, respectively. A user may first establish valid sets credentials and respective scope in initializing the PWAA, which is specific to the implementation. We provide a GUI to our reference implementation that salts and hashes a password provided to the user and retain an associated field to be populated with a token once initial connectivity is established. An archive with which a user has setup as private consults the PWAA for whether a supplied token is valid. If so, access is permitted. If no token or an invalid token is supplied, as described in Section 7.3.4 and the abstracted procedure in Figure 70 (Chapter 7).

8.3.3 STARGATE

A StarGate, as defined in Section 7.1.3, requires an extensible method of interaction with an open-ended set of Web services. As an exhibition of the three classes of attributes we described in Section 6.2, we have provided a reference implementation[6] to negotiate on the dimension of status code (a content-based attribute), Memento Damage [51] (a derived attribute), and a privacy boolean (an access attribute).

We leveraged a similar approach as MemGator and the MMA reference implementation in referring to an external configuration file as a basis. With MMAs, the sources were archival endpoints; with a StarGate, the configurations represent the endpoints and references to parsing algorithms. In our reference implementation of a StarGate, we defined the configuration using YAML (Figure 92). This configuration provides scoping of each service through indention, as inherent in YAML. A service key/descriptor is defined in the broadest scope with attributes of the service in the scoped indention. For the `damage` service, as shown in Figure 92, the expected return type is a `float`. A URI for the endpoint as well as method as supplying parameters to the endpoint (in this case `append` a URI) is also provided as well as a secondary method of parsing out the desired value from the response returned from the Damage API. The parser, here, is co-hosted with the StarGate (as indicated by the preceding "/" for the `parsing` URI) and the method of transferring that data to the parser (HTTP POST) is provided; in this case, the JSON returned from the Damage service is passed to the parser, which returns a float. It is the responsibility of

[6]`https://github.com/machawk1/stargate`

```
damage:
  type: float
  endpoint: http://memento-damage.cs.odu.edu/api/damage/
  endpointParameterization: append
  endpointType: string
  parsing: /parser/damage/
  parserParameterization: POST

statusCode:
  type: string
  endpoint: /service/status/
  endpointParameterization: append
  endpointType: string

privacy:
  type: json
  endpoint: /service/privacy/
  endpointParameterization: Prefer
  endpointType: json
```

Fig. 92 A sample StarGate configuration specifying expected parameters, resulting types, endpoints, and parsing URIs for content-based (`statusCode`), derived (`damage`), and access (`privacy`) attributes.

```
archive.alice.org private
archive.alice.org/collection/youtube/ public
carol.net/web/archive private
web.archive.org public
```

Fig. 93 A rudimentary boolean privacy classification service uses this data set to determine whether an archive is public or private.

the StarGate to then filter the results from these services to meet the requirements of the request, as expressed in the `Prefer` header.

Negotiation on Access Attributes

As a contribution toward aggregating private and public Web archives, we implemented the logic in the StarGate to exhibit the behavior described in Section 7.2.2 wherein a client/user-agent, at the request of a user, request captures that meet the criteria of Equations 4, 5, 6, 7, and 8 (Chapter 7) relating to requesting captures from permutations of private or public Web archives solely and through a precedence declaration, as described in that section. Attributing whether an archive is simply `public` or `private` can be explicitly configured with the StarGate or this boolean status inferred from the respective archive's holdings. It is likely that real-world scenarios require further class attribute to an archive beyond `public` or `private`. Inferring this attribute or automatically classifying an archive (keeping in-mind that a collection of holdings may have mixed as in Section 7.2.2) is beyond the scope of this dissertation (Section 9.2). We initially implemented the former approach of pre-configuring the classification service accessed by the StarGate (per Figure 92) with this privacy value per archive (inclusive of per-collection within an archive) with an assumption that an archive is `public` if there is no explicitly attribution as to its privacy.

Figure 93 shows a rudimentary definition for archival privacy to be used by the privacy classification endpoint at `/service/privacy/` (per Figure 92). The logic in the service is based off of the URI of the archive but may be more sophisticated for applications to be explored beyond this dissertation. When a StarGate is sent a request using `Prefer` of `privateOnly` (Section 7.3.6) with an additional `Prefer` header of the archival specification (encoded per Section 6.1.2), it sends a request to the classification service. This service accepts JSON using a similar `Prefer` mechanism and returns a JSON response, per

Figure 92. This process may be performed iteratively, though a StarGate may also use a services definition where an archival specification file is passed as a parameter and modified by the service, depending on the service's capability. In this latter case, the `endpointType` might be `json` and the `endpointParameterization` method might be different than `append` (e.g., passed via HTTP POST), as with the Damage `parser` in Figure 92.

Negotiation on Content-Based Attributes

Surfacing content-based attributes about a URI-M requires dereferencing the capture and extracting the relevant content. In the reference implementation, we setup a Web service endpoint that takes a URI-M as a parameter, queries the URI-M, and returns the status code (Appendix D). The purpose of using a separate Web service instead of coupling the StarGate with this functionality is to allow the StarGate to be functionally cohesive and extensible. A StarGate may send multiple requests, each with URI-M from a TimeMap to the service, and retain the response codes for caching purposes. An MMA communicating with the StarGate may obtain the status of this process (which may take time due to request throttling) using the HTTP status update pattern described in Section 7.2.1. Given the ultimate task of the StarGate is to filter on this list of URI-M-code pairs based on the client preferences as relayed through an MMA (or directly if a client supplied a TimeMap), the StarMap returned will include the `Preference-Applied` HTTP response header when the preference has been fully applied, i.e., all URI-Ms' HTTP status codes have an associated value as dereferenced.

Negotiation on Derived Attributes

Derived attributes require calculation beyond the process of access-and-parsing described in Section 8.3.3 but can exhibit a similar HTTP status update pattern as described in Section 7.2.1. We added an endpoint specification to be consumed by the StarGate base implementation that acquires the Memento Damage [51] value as calculated by the implemented service at `http://memento-damage.cs.odu.edu`. A StarGate similarly executes the process of awaiting a value to be generated by a service. Acquiring this `damage` score from the service requires more time to process than content-based attributes due to the inherent latency of the calculation and exhibited in accessing the API endpoint for the service. This service requires that a URI-R or URI-M be encoded and appended onto the URI `http://memento-damage.cs.odu.edu/api/damage/`. The service returns an extensive JSON response, for which we have provided an abbreviated example for a `cnn.com` URI-M[7] (prior to the URI-R becoming unarchivable [30]) in Appendix C.

Following receipt of this value, a StarGate will filter results similarly to doing so with the content-based attributes based on the preference relayed from the MMA or provided by a client directly through `Prefer`. The filtering process (e.g., Figure 91) is performed much in the same way as an MMA if performed by a StarGate. This interaction pattern for an MMA interacting with a StarGate exhibits MMAD_5.

8.3.4 $\text{MINK}^{\text{MINK}^{\text{MINK}}}$

In development of this dissertation, we created and publicly deployed Mink, a browser extension for Google Chrome that integrates and makes accessible Web archives when viewing the live Web, among other capabilities as detailed in Section 4.4. As previously indicated, interaction with Memento aggregators is often limited to the Time Travel service interface or on the command-line for users that are comfortable with CLIs. Mink provided an alternate means of interacting with Web archives using the Memento aggregation access pattern (Section 7.3.2) without having a destination site (e.g., the Time Travel service) or needing to use the command-line. It did so while still querying a Web accessible Memento aggregator for a static set of archives with the URL in the address bar being the URI-R to query the Web archives.

Our familiarity with the codebase (i.e., we created the tool) allowed for the exploration of enabling aggregation qua the Mementity Framework. The sole client-side software solution for Memento aggregation prior to this dissertation was MemGator, which required command-line access and either a one-off or client-side querying model, which might be intimidating to some Web archive users.

Mink queries an aggregator but browsers can readily query Web archives' Memento endpoints, albeit the resulting TimeMap is intended to be machine-readable. While requesting captures from individual archives would work in-place of an aggregator, but on an individual level, the logic for aggregation can be moved from the aggregator to Mink. In addition to the conventional querying model of pseudo-parallel querying of archives (Nomenclature N_2), which is inherently more efficient than allowing for querying precedence (Nomenclature N_3

[7]`http://memento-damage.cs.odu.edu/api/damage/http://web.archive.org/web/20161101131540/http://www.cnn.com/`

and Section 7.2.2), controlling the interfacing and allowing for either model to be used is exhibited in our improvements to Mink.

To accomplish this, our design considerations and deliverables (**MinkD**) consisted of interface patterns for grouping sources for:

MinkD$_1$: Simultaneous querying of archival subsets (Section 6.1)

MinkD$_2$: Defining order for querying (Section 7.2.2)

MinkD$_3$: Specifying additional archival sources (Section 6.1.2)

MinkD$_4$: Specifying additional parameters for querying personal and private archives (Section 7.3.4)

MinkD$_5$: The ability to easily and initiatively share querying preferences for collaboration and reproducibility (Section 7.2.4)

These aspects as applicable to the implementation of the framework at a high level are discussed in the following subsections.

Client-Side Archival Specification

We expanded Mink to be adaptable to a variety of expectations from an aggregator. Previously, it simply requested the URI currently being viewed in the browser from the ODUCS MemGator instance as the URI-R. With a more capable aggregator endpoint (i.e., an MMA), Mink can specify which archives it wants queried as well as the query precedence and short-circuiting models to be used. Upon customizing a set and order of archives in the interface displayed in Figure 94, a JSON version of the selection is generated and base64 encoded. On subsequent requests following the selection, Mink sends this value within the `Prefer` request header, as described in Section 8.3.1. Detecting the capability of the aggregator can be done by requesting information from the `/archives/about/` endpoint, as illustrated in Figure 55 (Chapter 6). This preliminary process is necessary to ensure the capability of the aggregator at the endpoint.

Client-Side Aggregator

While initially formulating this dissertation, we programmed the Mink base implementation [122] to perform a series of tasks:

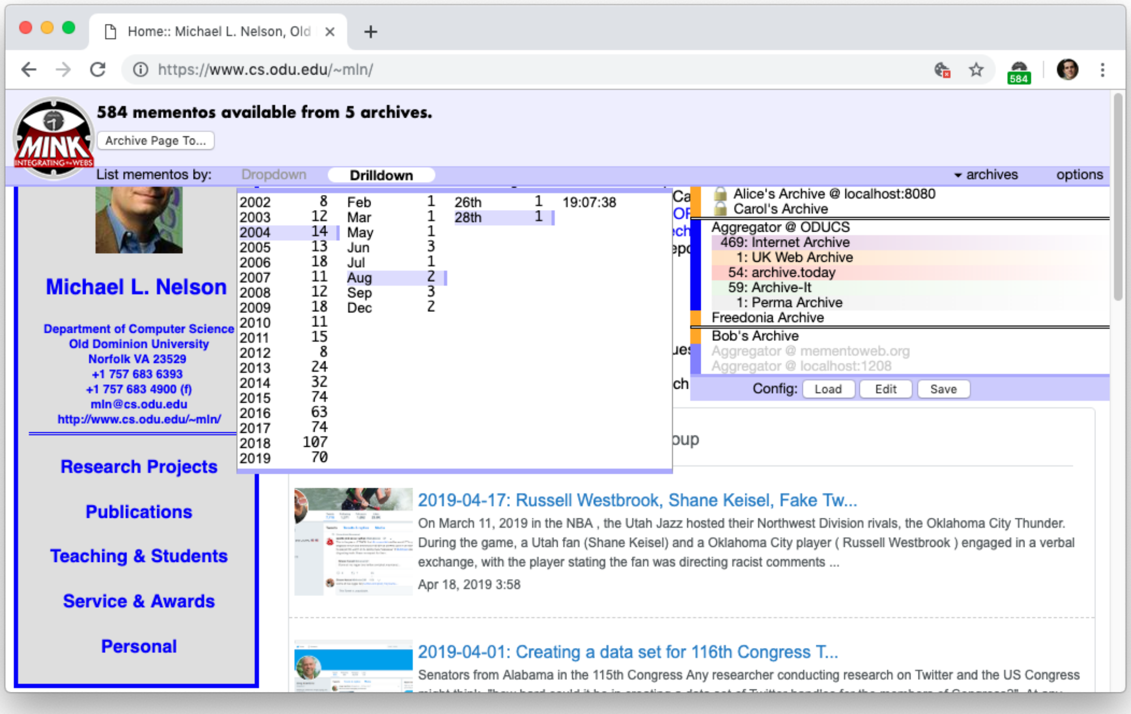
- Communicate with an aggregator to obtain a set of URI-Ms for the currently viewed URI-R.
- Allow the user to easily submit the currently viewed URI-R to a set of public Web archives.

In its original incarnation, Mink consulted the aggregator at `mementoweb.org` but upon changing the form in which TimeMaps are served (Section 4.4) and the creation of MemGator (Section 3.1.2), we changed Mink to consult a local instance of MemGator at Old Dominion University's Department of Computer Science[8]. We added the capability in Mink to display (on-demand within the UI) the per-archive breakdown of the captures (Figure 94) represented in the count indicator as well as the archival sources to be drag-and-dropped, imported from a URI, loaded from a local file, and saved for further sharing. We also provided user interface elements to share this configuration using IPFS (Section 8.3.4). Double clicking an archive in the interface disables it, preventing the respective archive from being queried while still retaining the endpoints for the archive in the configuration. The mapping of archive-to-name is performed on-the-fly based on a preset corpus of archives of which Mink is aware. If mementos from an unfamiliar hostname are returned, the archive is represented by the hostname. The color choices of each archived are derived from an extensive set of non-clashing, easily distinguishable colors per ColorBrewer[9] [87].

Client-to-Client Aggregation

With designing and creating InterPlanetary Wayback (Section 4.5), we introduced a novel way of sharing personal Web archives, facilitated by users sharing an index file (Section 7.2.4). When a user pushes their locally stored WARCs into IPFS using ipwb, a CDXJ index representative of the context of each resource representation in the WARC (e.g., URI-R, datetime) as well as content-addressed identifiers for the HTTP headers and entity are stored for future retrieval. Alice may share this CDXJ file with Carol so that Carol can replay Alice's captures without having Alice transfer the WARCs in a one-off procedure. Alice can also share the CDXJ file with Bob, who can do the same, even once Alice goes offline, so long as the captures still reside in IPFS. This latter point does necessarily require that Alice or Carol be the peer to share with Bob, as other peers allow for the nature of IPFS to be exhibited without a user having to explicitly specify peer with whom their holdings should be shared.

[8]`https://memgator.cs.odu.edu`
[9]`http://colorbrewer2.org`



```
2002     8   Feb   1   26th   1   19:07:38
2003    12   Mar   1   28th   1
2004    14   May   1
2005    13   Jun   3
2006    18   Jul   1
2007    11   Aug   2
2008    12   Sep   3
2009    18   Dec   2
2010    11
2011    15
2012     8
2013    24
2014    32
2015    74
2016    63
2017    74
2018   107
2019    70
```

Alice's Archive @ localhost:8080
Carol's Archive
Aggregator @ ODUCS
469: Internet Archive
1: UK Web Archive
54: archive.today
59: Archive-It
1: Perma Archive
Freedonia Archive
Bob's Archive
Aggregator @ mementoweb.org
Aggregator @ localhost:1208

Michael L. Nelson

Department of Computer Science
Old Dominion University
Norfolk VA 23529
+1 757 683 6393
+1 757 683 4900 (f)
mln@cs.odu.edu
http://www.cs.odu.edu/~mln/

2019-04-17: Russell Westbrook, Shane Keisel, Fake Tw...
On March 11, 2019 in the NBA , the Utah Jazz hosted their Northwest Division rivals, the Oklahoma City Thunder. During the game, a Utah fan (Shane Keisel) and a Oklahoma City player (Russell Westbrook) engaged in a verbal exchange, with the player stating the fan was directing racist comments ...
Apr 18, 2019 3:58

2019-04-01: Creating a data set for 116th Congress T...
Senators from Alabama in the 115th Congress Any researcher conducting research on Twitter and the US Congress

Fig. 94 Mink allows for the list of archives queried in the aggregation process to be displayed as last aggregated and customized.

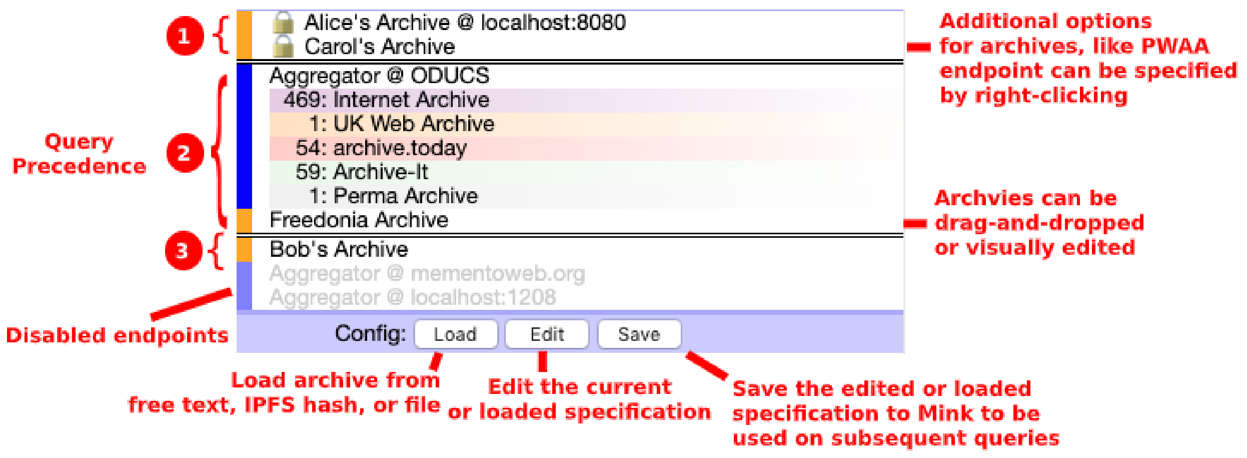

1
Alice's Archive @ localhost:8080
Carol's Archive
Additional options for archives, like PWAA endpoint can be specified by right-clicking
Query Precedence
2
Aggregator @ ODUCS
469: Internet Archive
1: UK Web Archive
54: archive.today
59: Archive-It
1: Perma Archive
Freedonia Archive
Archvies can be drag-and-dropped or visually edited
3
Bob's Archive
Aggregator @ mementoweb.org
Aggregator @ localhost:1208
Disabled endpoints
Config:
Load
Edit
Save
Load archive from free text, IPFS hash, or file
Edit the current or loaded specification
Save the edited or loaded specification to Mink to be used on subsequent queries

Fig. 95 In addition to specifying the distribution among Web archives, the improved Mink interface allows a user to import and export the configuration from a file, HTML `textarea`, or using IPFS hashes. A user can also visually or specify precedence separators and drag-and-drop archives to exhibit precedence on the generated and referenced specification within Mink.

While ipwb introduced new patterns for collaboration of personal Web archives, the Mementity Framework deals more with the aggregation aspect (i.e., TimeMaps) than on a memento-level like ipwb. The IPFS community has worked to implement an IPFS client in JavaScript [163], which we have leveraged in Mink to allow for the aforementioned archival specifications to be shared. Similar to how users share CDXJ index files in ipwb (Figure 75 in Chapter 7), archival specifications may be pushed to IPFS to create an IPFS hash that would allow users to share archives. For example, Alice has implicitly generated an archival specification in Mink by specifying which archives she wants aggregated and in which order. She wants others to be able to reuse her curated list of archives and customizations for querying, so may “Share” the configuration within Mink (Figure 95). This causes the specification to be added to IPFS using the JavaScript-based IPFS daemon provided by `js-ipfs` and embedded in Mink. Retrieval of this archival specification then requires only sharing this relatively short hash. Once Alice has shared this hash with Carol, Carol’s Mink instance will either attempt to aggregate based on the specification or consult Alice’s Mink instance (if online) for further, updated aggregation procedure.

8.4 HOW WELL ARE CHAPTER 1 SCENARIOS AND ACCESS PATTERNS REALIZED

We evaluated the level of resolution of the scenarios and issues described in Chapter 1. With the mementities of the Mementity Framework in-place, there was still be a question of ease-of-use with the reference implementations when they implement all features of the framework. This influenced the user interface decisions for integration to encourage the adoption of the framework through a user experience with minimal barriers.

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 55), 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 81 in Chapter 7). 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 (Chapter 1) showed two captures of the same URI, one of a personal and private representation and the other of a generic login page. Figure 7 (Chapter 1) shows multiple captures of `cnn.com` of a variety of qualities. By surfacing these attributes (privacy and damage, respectively) and other dimensions through the Memento extension of StarMaps (Section 6) 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 7.1.3), these scenarios in Chapter 1 may be considered resolved with the application of the Mementity Framework.

It Was There Yesterday, Where Did It Go?

Given the live Web is ephemeral and institutional Web archives often miss much of the live Web content, it is understandable that content individuals care about is lost in time due to the technical capability of the tools, among other reasons. In Chapter 4 we created tools to mitigate this problem for the creation of Web archives using browser-based preservation (Section 4.2) and personal Web archival replay systems

(Sections 4.3 and 4.5). The latter facilitates permanence of personal Web archives beyond preservation with the potential for distributing the bearer role while also doing so securely using encryption.

Save This, But Only For Me:
Archival access is a central theme in this dissertation. Through the introduction of the Private Web Archive Adapter (Section 7.1.2), the ability for personal and private Web archiving as above, and the user access patterns in Section 7.3, an individual can preserve live Web content and regulate access to it.

I Want to Share This But Control Who Can See It
Collaboration of Web archives facilitates a more comprehensive picture of the *Web that was* (e.g., Figure 53 in Chapter 6). Personalization of Web pages behind that originally resided behind authentication on the live Web can have significantly different representations from each other and that which is captured by public Web archives. Sharing captures in the form of the mementos themselves (as done by ipwb in Section 4.5), URI-Ms (as exhibited by ipwb's collaboration and propagation model in Section 7.2.4), or TimeMaps of private URI-Ms securely requires access regulation to each of these sources of private mementos. While we above described access regulation using PWAA, the aggregation access patterns that exhibits the picture of the *Web that was* instead of piecemeal mementos (i.e., a URI-R over time) require negotiation in the privacy dimension. By surfacing this attribute of captures using Access Attributes (Section 6.2.3) and representing the status, upon authentication within aggregated StarMaps (Figure 64 in Chapter 6), a user can share their private captures but also control who can see it.

8.5 SUMMARY

In this chapter we evaluated the Mementity Framework. We justified the design decisions of the mementities' scope and roles in Section 8.1. In Section 8.2 we abstracted the costs incurred when implementing the Mementity framework in a Web archiving workflow. Section 8.3 described the implementation of the framework inclusive of concrete programmatic products that exhibit deliverables from the justified design decisions in Section 8.1. In Section 8.3 we also extended on the core framework implementations onto a browser extension to emphasize its applicability beyond the use cases described in this dissertation. In

Section 8.4 we circled back to the scenarios in Chapter 1 and how the Mementity Framework addresses each.

CHAPTER 9

CONTRIBUTIONS, CONCLUSIONS, AND FUTURE WORK

Just because nothing shakes loose from the web, doesn't mean the spider went hungry.

- D.M. Timney

In this chapter we provide a review of Research Questions 1-6 and how each was addressed in this dissertation. In Section 9.1 we enumerate our contributions introducing and further exploring the nuances of the Mementity Framework. Section 9.2 describes future work beyond the scope but facilitated by this dissertation. In Section 9.3, we summarize our conclusions in exploring the research described in this dissertation. The remainder of this section details how we addressed the research questions.

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

In Chapter 5, we described our studies on archivability. Here we identified that JavaScript was one of the primary culprits affecting archival quality. These studies did not take into account content behind authentication, which was difficult to preserve prior to the work performed in this dissertation. While content behind authentication can now be preserved (Section 4.2), access control is required for these captures, which may contain sensitive content. We introduced and implemented symmetric encryption for replay within InterPlanetary Wayback (Section 4.5). We also introduced a technical means of regulating access (Section 7.1.2) to ease and make more systematic the replay procedure for this sort of content.

To meet this requirement we published multiple peer-reviewed papers inclusive of those describing tools like WARCreate [125], ArchiveNow [23], and Unobtrusive Replay Banners [9]. We also published multiple studies identifying content that is difficult to capture and replay with the Archival Acid Test [123], the Change of Archivability Over Time [118], the Impact of JavaScript on Archivability [52], and Measuring the Impact of Missing Resources [50, 51].

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

Through our archivability studies, we highlighted that relying on tools for preservation that are not based on the means of viewing the Web (i.e., not a Web browser) is a root cause for degraded archival quality (Memento Damage [51]). We evaluated and compared these tools through an Archival Acid Test (Section 5.3). This provided a basis for the capability of archival Web crawlers that did not leverage a Web browser for preservation. We initially leveraged the browser extension APIs with WARCreate (Section 4.2) to allow for content that was inaccessible to archival crawlers to be archived. We extended this concept by leveraging a headless browser within an archival crawling procedure with WAIL-Electron (Section 4.3.2) to facilitate high fidelity archiving of content behind authentication at a larger scale.

In explorations of leveraging browser APIs and evaluating crawlers versus preservation with native tools, we published multiple peer-reviewed works including the aforementioned WARCreate [125], Mink [122], Mobile Mink [106], and WAIL [32]. We also performed peer-reviewed published studies on Identifying Personalized Representations in the Archive [117], using Service Workers for Archival Replay [8], and the Unobtrusive and Extensible Archival Replay Banners [9].

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

Content behind authentication is more difficult to preserve due to it inherently not being publicly accessible. The tools we created and described in Chapter 4 mitigated this issue. WARCreate (Section 4.2) leveraged the browser extension APIs to circumvent the restriction of needing to supply credentials by having access to the pages behind authentication directly for preservation. We highlighted the need for access regulation for private captures and provided a solution to this issue for replaying content behind authentication with the framework's Private Web Archive Adapter (Section 7.1.2).

For this Research Question we published two distinct peer-review papers, the initial being an extensive study of measuring the impact of URI canonicalization [115, 116], which informed our use case for content-based attributes (Section 6.2.1). The second paper consisted of an explanation of the core concepts of the Mementity Framework [124] and was peer reviewed. The reviews to the latter publication helped to further inform the work of this dissertation.

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

In Chapter 7 we introduced the Mementity Framework, which leverages standards to indicate that aggregated URI-Ms representing personal and private Web archives, potentially behind authentication, require special handling. Our proposed method of content negotiation in dimensions beyond time in Chapter 6 allows for a systematic procedure for identifying archives that require this process through the expression of access attributes (Section 6.2.3) in CDXJ TimeMaps.

We had three peer-reviewed publications related to this Research Question. The first related to the initial publishing of InterPlanetary Wayback [7] with the second integrating encryption, privacy aspects, and further evaluation [113]. The third peer-reviewed publication consisted of our abbreviated description of the concepts in the Mementity Framework [124].

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?

The access attributes described in Section 6.2.3 provide a means for aggregators to express that accessing private content requires special handling beyond simply dereferencing the URI-M. These attributes allow for captures that are aggregated between personal, private, and public Web archives to indicate that a subsequent authentication procedure may be needed prior to access.

We published two peer-reviewed paper relating to archival access to address this research question. The first was the initial publication of Mink [122], which was a novel means of integrating the live and archived Web but additionally allowed client-side aggregation of these with local Web archives on the user's machine. The second peer-reviewed publication relating to this Research Question resided in our aforementioned abbreviated version of the Mementity Framework [124].

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

In Section 7.1.2 we introduced the Private Web Archive Adapter mementity that implemented an OAuth 2-based tokenization mechanism to regulate access. While this method was provided in the reference implementation (Section 8.3.2), the Mementity Framework is extensible to allow for interoperability of access methods beyond those described in this dissertation. Alluding to standards has facilitated this extensibility of the framework (Section 7.4) for future methods of access control.

Our publications for RQ4 also took into account the challenges of RQ6 in that both of the peer-reviewed ipwb publications [7, 113] as well as the abbreviated version of the Mementity Framework [124] investigated access control as is relevant for private Web archives. In total, we published 17 papers while investigating the research contained in this dissertation.

9.1 CONTRIBUTIONS

This dissertation contributes to the integration of personal and private Web archives with public Web archives. While much prior research in Web archiving has been devoted to preservation of the public live Web and publicly replaying these captures, the contributions of this dissertation are novel on multiple fronts. Our focus has been on supplementing archival interaction by extending Memento, which formally introduced the notion of time on the Web. The concepts and contributions in this dissertation are applicable to the archived Web beyond negotiation in time. Our contributions are as follows:

1. We provided a hierarchical approach at supplementing the set of Web archives aggregated using the "Memento Meta-Aggregator" abstraction.

2. We introduced a standard, extensible mechanism for regulating access to private Web archives using the "Private Web Archive Adapter" abstraction.

3. We leveraged a standard mechanism in HTTP `Prefer` to enable clients to specify which archives are aggregated along with the parameters of their typical query.

4. We created a novel approach to preserving content behind authentication by leveraging Web browser extension APIs to allow content that was previously inaccessible to archival crawlers to be preserved in the standard WARC format.

5. We integrated the live and archived Web viewing experience through Mink, a means for clients to view how well archived a live Web URI is captured in terms of quantity based on integration with a remote Memento aggregator.

6. We extended Mink to allow client-side aggregation from a browser where aggregation is normally limited to querying a server or querying archives directly and being performed the procedure manually.

7. We integrated the InterPlanetary File System with the WARC standard via InterPlanetary Wayback to allow for client-side archival replay without needing to possess the archival holdings.

8. We enabled encryption and decryption of personal archival holdings within InterPlanetary Wayback.

9.2 FUTURE WORK

While this work focused on the aggregation of private and personal Web archives with public Web archives, the latter of which is typically the sole scope of focus in conventional Web archive research, there remains a large amount of research that can be investigated beyond and as facilitated by this dissertation. For example, in Section 6.1.2 we leverage HTTP `Prefer` with the `archives` preference and a data URI descriptor for client-side archival specification. The choice of this keyword allowed us to leverage the Prefer standard while reducing the potential for a name clash that would occur from using a preference like `config` in lieu of `archives`. However, in doing this, we were required to separate out negotiation in other dimensions beyond time (Section 7.2.1) into separate `Prefer` headers (allowed by the Prefer specification) instead of combining multiple preferences into a single, comprehensive expression. Part of this reason is due to the lack of semantics and syntax for expressing ranges and thresholds. This is exhibited in the need for awkwardly declaring `Prefer: damage="<0.5"` in Figure 71 (Chapter 7) instead of a more syntactically natural `Prefer: damage<0.5`, which is disallowed by the specification and HTTP in general.

The Mementity Framework also dealt with Web archives on the level of aggregation when there still remains the issue of potential leakage and privacy violations between aggregated private and public Web archives. In Chapter 8 we describe a scenario where the privacy of an archive can be inferred instead of explicitly specified, which might be a complicated classification process that was beyond the scope of this dissertation.

Because the majority of Web archiving research focuses on the public live Web and public archived Web, there is little research beyond this dissertation for studies on the private archived Web. Through aggregation, these private and personal captures become more standard and discoverable, which hopefully facilitates further studies of private Web archives.

9.3 CONCLUSIONS

This work introduced and explored the Mementity Framework – a framework for aggregating private and public Web archives. Through preliminary investigations of Web archives, we found that the public live Web is not comprehensively archived, even by contemporary tools, and the public live Web is often neglected by archiving institutions. The Mementity Framework introduced three components (mementities), the Memento Meta-Aggregator, the Private Web Archive Adapter, and the StarGate, into the Web archiving work flow to extend on conventional practice and account for nuances of including private Web archives into the historical record as represented by Web archives.

The framework provided systematic solutions to the scenarios in Chapter 1. Content behind authentication like online bank statements (Figure 5) and born-digital photos (Figure 1) can be easily preserved from a browser, despite residing behind authentication on the live Web. Regulating access to these captures can be accomplished using the PWAA in Chapter 7. Propagation of these captures, in the case of securely sharing baby photos, can be accomplished using tools described in Chapter 4, namely InterPlanetary Wayback. Aggregating photos among those a user chooses instead of done so public can be accomplished using the three mementities (Section 7.1) for richer Memento aggregation (via the MMA), systematic access control of the private captures (via the PWAA), and negotiation in the dimensions of privacy (via the StarGate). These online captures can be replayed in Web Archiving Integration Layer (Section 4.3), as the preservation format reused by this dissertation, among other formats and use cases embedded into the Mementity Framework, reuses established standards. Though we provided initial use cases and access patterns for the Mementity Framework (Section 7.3), it is inherently extensible (Section 7.4) because of the initial design decisions (Section 8.1) for mementity cohesiveness and interoperability. This dissertation helps to mitigate the issue of ephemerality on the often unpreserved private live Web (Section 1.2) by facilitating personal and private preservation of the Web (Chapter 4). The Mementity Framework further facilitates permanence of this content by providing the ability to regulate access to these captures (Section 1.3) and encouraging collaboration to give a more comprehensive of the complete Web (not just the public) that was.

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] Abrams, D., Baecker, R., and Chignell, M. Information Archiving with Bookmarks: Personal Web Space Construction and Archiving. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1998), pp. 41–48.

[4] Ainsworth, S. G., Alsum, A., SalahEldeen, H., Weigle, M. C., and Nelson, M. L. How Much of the Web is Archived? In *Proceeding of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)* (2011), pp. 133–136.

[5] Alabbas, A., and Bell, J. Indexed Database API 2.0. *W3C* (2018). https://www.w3.org/TR/IndexedDB/.

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

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

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

[9] Alam, S., Kelly, M., Weigle, M. C., and Nelson, M. L. Unobtrusive and Extensible Archival Replay Banners Using Custom Elements. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)* (2018), pp. 319–320.

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

[11] Alam, S., Nelson, M. L., de Sompel, H. V., and Rosenthal, D. S. H. Web Archive Profiling Through Fulltext Search. In *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL)* (2016), pp. 121–132.

[12] Alam, S., Nelson, M. L., Van de Sompel, H., Balakireva, L. L., Shankar, H., and Rosenthal, D. S. H. Web archive profiling through CDX summarization. *International Journal on Digital Libraries 17*, 3 (2016), 223–238.

[13] AlNoamany, Y., AlSum, A., Weigle, M. C., and Nelson, M. L. Who and What Links to the Internet Archive. In *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL)* (2013), pp. 346–357.

[14] AlNoamany, Y., AlSum, A., Weigle, M. C., and Nelson, M. L. Who and What Links to the Internet Archive. *International Journal on Digital Libraries 14*, 3-4 (2014), 101–115.

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

[16] AlSum, A., and Nelson, M. L. Thumbnail Summarization Techniques for Web Archives. In *Proceedings of the 36th European Conference on IR Research (ECIR 2014)* (2014), pp. 299–310.

[17] AlSum, A., Weigle, M. C., Nelson, M. L., and de Sompel, H. V. Profiling Web Archive Coverage for Top-Level Domain and Content Language . In *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL)* (2013), pp. 60–71.

[18] AlSum, A., Weigle, M. C., Nelson, M. L., and Van de Sompel, H. Profiling Web Archive Coverage for Top-Level Domain and Content Language. *International Journal on Digital Libraries 14*, 3-4 (2014), 149–166.

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

[20] Antoniades, D., Polakis, I., Kontaxis, G., Athanasopoulos, E., Ioannidis, S., Markatos, E. P., and Karagiannis, T. we.b: The web of short URLs. In *Proceedings of the 20th International Conference on World Wide Web* (2011), pp. 715–724.

[21] Asimov, I. *Pebble in the Sky.* Doubleday, Garden City, New York, 1950.

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

[23] Aturban, M., Kelly, M., Alam, S., Berlin, J. A., Nelson, M. L., and Weigle, M. C. ArchiveNow: Simplified, Extensible, Multi-Archive Preservation. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)* (2018), pp. 321–322.

[24] Bailey, J., Grotke, A., Hanna, K., Hartman, C., McCain, E., Moffatt, C., and Taylor, N. Web Archiving in the United States: A 2013 Survey. Tech. rep., The National Digital Stewardship Alliance. `http://ndsa.org/documents/NDSA_USWebArchivingSurvey_2013.pdf`.

[25] Bailey, J., Grotke, A., McCain, E., Moffatt, C., and Taylor, N. Web Archiving in the United States: A 2016 Survey. Tech. rep., The National Digital Stewardship Alliance. `http://ndsa.org/documents/WebArchivingintheUnitedStates_A2016Survey.pdf`.

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

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

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

[29] Benet, J. IPFS - Content Addressed, Version, P2P File System. Tech. Rep. arXiv:1407.3561, July 2014.

[30] Berlin, J. 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.

[31] Berlin, J. 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.

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

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

[34] Berners-Lee, T. Web Architecture: Generic Resources. `http://www.w3.org/DesignIssues/Generic.html`.

[35] Berners-Lee, T. Cool URIs don't change. `http://www.w3.org/Provider/Style/URI.html`.

[36] Berners-Lee, T. Universal Resource Identifiers – Axioms of Web Architecture. `https://www.w3.org/DesignIssues/Axioms.html`.

[37] Berners-Lee, T. What do HTTP URIs Identify? `https://www.w3.org/DesignIssues/HTTP-URI.html`.

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

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

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

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

[42] Bornand, N. J., Balakireva, L., and Van de Sompel, H. Routing Memento Requests Using Binary Classifiers. In *Proceedings of the ACM/IEEE-CS on Joint Conference on Digital Libraries (JCDL)* (2016), pp. 63–72.

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

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

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

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

[47] 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.

[48] Brunelle, J. F. *Scripts in a Frame: A Framework for Archiving Deferred Representations.* PhD thesis, Old Dominion University, Norfolk, Virginia, USA, 2016.

[49] Brunelle, J. F., Ferrante, K., Wilczek, E., Weigle, M. C., and Nelson, M. L. Leveraging Heritrix and the Wayback Machine on a Corporate Intranet: A Case Study on Improving Corporate Archives. *D-Lib Magazine 22*, 1/2 (2016).

[50] Brunelle, J. F., Kelly, M., SalahEldeen, H., Weigle, M. C., and Nelson, M. L. 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)* (2014), pp. 321–330.

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

[52] Brunelle, J. F., Kelly, M., Weigle, M. C., and Nelson, M. L. The Impact of JavaScript on Archivability. *International Journal on Digital Libraries 17*, 2 (2016), 95–117.

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

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

[55] Burner, M., and Kahle, B. Arc File Format. `http://archive.org/web/researcher/ArcFileFormat.php`, September 1996.

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

[57] Chisholm, W., Vanderheiden, G., and Jacobs, I. Web Content Accessibility Guidelines 1.0. *Interactions 8*, 4 (July 2001), 35–54.

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

[59] Crocker, D., and Overell, P. Augmented BNF for Syntax Specifications: ABNF. IETF RFC 5234, January 2008.

[60] Day, M. Collecting and preserving the World Wide Web. Tech. rep., Joint Information Systems Committee (JISC), 2003. `http://library.wellcome.ac.uk/assets/WTL039229.pdf`.

[61] Denis Petrov. Please do not use http://archive.IS mirror for linking, use others mirrors [.TODAY .FO .LI .VN .MD .PH]. .IS might stop working soon. `https://twitter.com/archiveis/status/1081276424781287427`, January 2019.

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

[63] Deveria, A. Can I use... Support tables for HTML5, CSS3, etc - Service Workers, 2018. `https://caniuse.com/#feat=serviceworker`.

[64] Dickey, T. E. Lynx – The Text Web-Browser. `http://lynx.invisible-island.net/`.

[65] (Director), G. L. Star Wars: Episode II - Attack of the Clones. 20th Century Fox, 2002.

[66] Donovan, A. A. A., and Kernighan, B. W. *The Go Programming Language*. Addison-Wesley Professional, 2015.

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

[68] Eastlake, D. E., and Jones, P. E. US Secure Hash Algorithm 1 (SHA1). IETF RFC 3174, September 2001.

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

[70] Ecma International. ECMA-262 6.0: Set Objects. `https://www.ecma-international.org/ecma-262/6.0/#sec-set-objects`, June 2015.

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

[72] Faulkner, W. *Requiem for a Nun.* Random House, New York, 1951.

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

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

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

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

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

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

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

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

[81] Goel, V. Web Archive Analysis. Presented at Archive-It Partner Meeting, November 2013. https://support.archive-it.org/hc/article_attachments/208693066/vinay-goel-ait-partner-meeting-slc-2013.pdf.

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

[83] Gomes, D., Freitas, S., and Silva, M. J. Design and Selection Criteria for a National Web Archive. In *Research and Advanced Technology for Digital Libraries*. Springer, 2006, pp. 196–207.

[84] Gomes, D., Miranda, J., and Costa, M. A Survey on Web Archiving Initiatives. In *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL)* (2011), pp. 408–420.

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

[86] Hardt, D. The OAuth 2.0 Authorization Framework. IETF RFC 6749, October 2012.

[87] Harrower, M., and Brewer, C. A. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal 40*, 1 (2003), 27–37.

[88] Hégaret, P. L., Whitmer, R., and Wood, L. Document Object Model, January 2005. https://www.w3.org/DOM/.

[89] Hickson, I. Acid Tests. http://www.acidtests.org/, 2013.

[90] Hickson, I., Berjon, R., Faulkner, S., Leithead, T., Navara, E. D., O'Connor, E., and Pfeiffer, S. Document Metadata — HTML5, October 2014.

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

[92] Huurdeman, H. C., Ben-David, A., and Sammar, T. Sprint Methods for Web Archive Research. In *Proceedings of the 5th Annual ACM Web Science Conference* (2013), pp. 182–190.

[93] Huxley, A. A. *The Perennial Philosophy.* Harper & Bros., New York City, 1944.

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

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

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

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

[98] ISO28500:2017. WARC (Web ARChive) file format. `https://www.iso.org/standard/68004.html`, August 2017.

[99] Jackson, A. N. The CDX file format (2015). `https://iipc.github.io/warc-specifications/specifications/cdx-format/cdx-2015/`, November 2015.

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

[101] Jackson, A. N. UKWA Documentation. `https://github.com/ukwa/ukwa-documentation/blob/master/README.md`, May 2018.

[102] Jacobs, I., and Walsh, N. Architecture of the world wide web, volume one. Tech. Rep. W3C Recommendation 15 December 2004, W3C, 2004.

[103] Jones, M. B., and Hardt, D. The OAuth 2.0 Authorization Framework: Bearer Token Usage. IETF RFC 6750, October 2012.

[104] Jones, S. M., Van de Sompel, H., and Nelson, M. L. Mementos in the Raw. `http://ws-dl.blogspot.com/2016/04/2016-04-27-mementos-in-raw.html`, April 2016.

[105] Jones, S. M., Van de Sompel, H., Shankar, H., Klein, M., Tobin, R., and Grover, C. Scholarly Context Adrift: Three out of Four URI References Lead to Changed Content. *PLOS ONE 11*, 12 (12 2016), 1–32.

[106] Jordan, W., Kelly, M., Brunelle, J. F., Vobrak, L., Weigle, M. C., and Nelson, M. L. Mobile Mink: Merging Mobile and Desktop Archived Webs. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)* (2015), pp. 243–244.

[107] Justin F. Brunelle. Zombies in the Archives. `http://ws-dl.blogspot.com/2012/10/2012-10-10-zombies-in-archives.html`, October 2012.

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

[109] Kelly, M. machawk1/archivalAcidTest: Evaluating Archival Crawlers and Tools. `https://github.com/machawk1/archivalAcidTest`, 2014.

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

[111] Kelly, M. 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`.

[112] Kelly, M. 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.

[113] Kelly, M., Alam, S., Nelson, M. L., and Weigle, M. C. InterPlanetary Wayback: Peer-To-Peer Permanence of Web Archives. In *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL)* (2016), pp. 411–416.

[114] Kelly, M., Alam, S., Nelson, M. L., and Weigle, M. C. Client-Assisted Memento Aggregation Using the Prefer Header. Presented at Web Archiving and Digital Libraries (WADL) Workshop, June 2018. https://www.slideshare.net/matkelly01/clientassisted-memento-aggregation-using-the-prefer-header.

[115] Kelly, M., Alkwai, L. M., Alam, S., Nelson, M. L., Weigle, M. C., and Van de Sompel, H. Impact of URI Canonicalization on Memento Count. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)* (2017), pp. 303–304.

[116] Kelly, M., Alkwai, L. M., Nelson, M. L., Weigle, M. C., and Van de Sompel, H. Impact of URI Canonicalization on Memento Count. Tech. Rep. arXiv:1703.03302, Old Dominion University, March 2017.

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

[118] Kelly, M., Brunelle, J. F., Weigle, M. C., and Nelson, M. L. On the Change in Archivability of Websites Over Time. In *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL)* (2013), pp. 35–47.

[119] Kelly, M., and Dias, D. 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.

[120] Kelly, M., Nelson, M. L., and Weigle, M. C. 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.

[121] Kelly, M., Nelson, M. L., and Weigle, M. C. 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.

[122] Kelly, M., Nelson, M. L., and Weigle, M. C. 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)* (2014), pp. 469–470.

[123] Kelly, M., Nelson, M. L., and Weigle, M. C. 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)* (2014), pp. 25–28.

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

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

[126] Kelly, M., Weigle, M. C., and Nelson, M. L. 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.

[127] Kelly, M., Weigle, M. C., and Nelson, M. L. 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.

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

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

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

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

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

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

[134] Lindley, S. E., Marshall, C. C., Banks, R., Sellen, A., and Regan, T. Rethinking the Web as a Personal Archive. In *Proceedings of the 22nd International Conference on World Wide Web* (2013), pp. 749–760.

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

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

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

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

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

[140] Masanès, J. Web Archiving: Issues and Methods. In *Web Archiving*. Springer, 2006, pp. 1–53.

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

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

[143] McGrath, J., Peaker, A., Cordell, R., and Dillon, E. M. Our Marathon: The Boston Bombing Digital Archive. http://marathon.neu.edu/, 2013-2015.

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

[145] Mohr, G., Stack, M., Ranitovic, I., Avery, D., and Kimpton, M. An Introduction to Heritrix, An open source archival quality web crawler. In *Proceedings of the 4th International Web Archiving Workshop (IWAW'04)* (September 2004).

[146] Munroe, R. xkcd: Dimensions. https://xkcd.com/1524/, May 2005.

[147] Murchison, K. Use of the Prefer Header Field in Web Distributed Authoring and Versioning (WebDAV). IETF RFC 8144, April 2017.

[148] NDSA Content Working Group. Web Archiving Survey Report. Tech. rep., The National Digital Stewardship Alliance. http://www.digitalpreservation.gov/documents/ndsa_web_archiving_survey_report_2012.pdf.

[149] Negulescu, K. C. Web archiving @ the internet archive. Presentation at the 2010 Digital Preservation Partner Meeting, http://www.digitalpreservation.gov/meetings/documents/ndiipp10/NDIIPP072110FinalIA.ppt, 2010.

[150] Nelson, M. L. Archive.is Supports Memento. https://ws-dl.blogspot.com/2013/07/2013-07-09-archiveis-supports-memento.html, July 2013.

[151] Nelson, M. L. Blockchain Can Not Be Used To Verify Replayed Archived Web Pages. Presented at the Coalition for Networked Information (CNI) Fall 2018 Membership Meeting, December 2018. https://www.slideshare.net/phonedude/blockchain-can-not-be-used-to-verify-replayed-archived-web-pages-125618706.

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

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

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

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

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

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

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

[159] Petrov, D. Archive.is blog — Curious... Why the move in domain names from... `https://blog.archive.today/post/82775187091/curious-why-the-move-in-domain-names-from`, April 2014.

[160] Phillips, M. E. PANDORA, Australia's Web Archive, and the Digital Archiving System that Supports It. *DigiCULT.info* (December 2003), 24.

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

[162] Potts, L., Seitzinger, J., Jones, D., and Harrison, A. Tweeting Disaster: Hashtag Constructions and Collisions. In *Proceedings of the 29th ACM International Conference on Design of Communication* (2011), pp. 235–240.

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

[164] Ranganathan, A., and Sicking, J. File API. *W3C* (2015). `https://www.w3.org/TR/FileAPI/`.

[165] Rao, H. C.-H., Chen, Y.-F., and Chen, M.-F. A Proxy-based Personal Web Archiving Service. *SIGOPS Operating System Review 35*, 1 (Jan. 2001), 61–72.

[166] Rappin, N., and Dunn, R. *wxPython in Action.* 2006.

[167] Rauber, A., Kaiser, M., and Wachter, B. Ethical Issues in Web Archive Creation and Usage-Towards a Research Agenda. In *Proceedings of the 8th International Web Archiving Workshop (IWAW'08)* (2008).

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

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

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

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

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

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

[174] Rosenthal, D. S. H., Robertson, T., Lipkis, T., Reich, V., and Morabito, S. Requirements for Digital Preservation Systems. *D-Lib Magazine 11*, 11 (November 2005).

[175] Rosenthal, D. S. H., Vargas, D. L., Lipkis, T. A., and Griffin, C. T. Enhancing the LOCKSS digital preservation technology. *D-Lib Magazine 21*, 9/10 (Sept/Oct 2015).

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

[177] Rothenberg, J. Ensuring the Longevity of Digital Information. *International Journal of Legal Information 26*, 1 (1998), 1–18.

[178] Ruderman, J. Same-origin policy - Web security. `https://developer.mozilla.org/en-US/docs/Web/JavaScript/Same_origin_policy_for_JavaScript`, March 2019.

[179] Russell, A., Song, J., Archibald, J., and Kruisselbrink, M. Service Workers. *W3C* (2017). `https://www.w3.org/TR/service-workers/`.

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

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

[182] Sauermann, L., and Cyganiak, R. Cool URIs for the semantic web. Tech. Rep. W3C Interest Group Note 31 March 2008, W3C, 2008.

[183] Schwarz, T., Baker, M., Bassi, S., Baumgart, B., Flagg, W., van Ingen, C., Joste, K., Manasse, M., and Shah, M. Disk Failure Investigations at the Internet Archive. In *Work-in-Progess session, NASA/IEEE Conference on Mass Storage Systems and Technologies (MSST2006)* (2006).

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

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

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

[187] Smart, J., Hock, K., and Csomor, S. *Cross-Platform GUI programming with wxWidgets*. 2005.

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

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

[190] Strodl, S., Motlik, F., Stadler, K., and Rauber, A. Personal & Soho Archiving. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)* (2008), pp. 115–123.

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

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

[193] Tofel, B. 'Wayback' for Accessing Web Archives. In *Proceedings of the 7th International Web Archiving Workshop (IWAW'07)* (2007).

[194] Tweedy, H., McCown, F., and Nelson, M. L. A Memento Web Browser for iOS. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)* (2013), pp. 371–372.

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

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

[197] Van de Sompel, H., Nelson, M. L., Balakireva, L., Klein, M., Jones, S. M., and Shankar, H. Mementos In the Raw, Take Two. `http://ws-dl.blogspot.com/2016/08/2016-08-15-mementos-in-raw-take-two.html`, August 2016.

[198] Van de Sompel andMichael Nelson, H., and Sanderson, R. HTTP Framework for Time-Based Access to Resource States – Memento. IETF RFC 7089, December 2013.

[199] Vukotic, A., and Goodwill, J. *Apache Tomcat 7*, 1st ed. Apress, 2011.

[200] (W3C), W. W. W. C. Web Accessibility Initiative (WAI). `https://www.w3.org/WAI`. Accessed: 2014-01-08.

[201] Web Hypertext Application Technology Work Group. The 2D rendering context. *WHATWG Living Standard* (2019). `https://html.spec.whatwg.org/multipage/canvas.html#2dcontext`.

[202] Web Hypertext Application Technology Work Group. Web storage. *WHATWG Living Standard* (2019). `https://html.spec.whatwg.org/multipage/webstorage.html#webstorage`.

[203] Webber, J. 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.

[204] Weigle, M. C., Nelson, M. L., and Potts, L. "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.

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

[206] Zibricky, M., Goebel, H., Cortesi, D., and Vierra, D. PyInstaller Quickstart - PyInstaller bundles Python applications. `https://www.pyinstaller.org/`, 2016.

APPENDIX A

ARCHIVES

There are many Web archiving efforts. This table documents a few of the currently existing public ones referenced throughout this dissertation.

Archive	URI	Memento Support
Internet Archive	`https://archive.org/web/`	Yes
UK Web Archive	`https://www.webarchive.org.uk`	Yes
WebCite	`http://www.webcitation.org`	No, Proxied
archive.is	`https://archive.is`	Yes
Archive-It	`https://archive-it.org`	Yes
Mummify.it	`http://mummify.it`	N/A, defunct
Perma.cc	`https://perma.cc`	Yes

APPENDIX B

SAMPLE ARCHIVAL SPECIFICATION

Memento aggregators internally use a configuration for a set of archives to be queried when a URI-R and optionally as datetime are requested from a client. This appendix provides an abbreviated example of a JSON-formatted aggregator configuration, as adapted from one provided online at `https://git.io/archives` by MemGator [10].

```
[
  {
    "id": "ia",
    "name": "Internet Archive",
    "timemap": "http://web.archive.org/web/timemap/link/",
    "timegate": "http://web.archive.org/web/",
  },
  {
    "id": "proni",
    "name": "PRONI Web Archive",
    "timemap": "http://webarchive.proni.gov.uk/timemap/",
    "timegate": "http://webarchive.proni.gov.uk/timegate/",
  },
  {
    "id": "pastpages",
    "name": "PastPages Web Archive",
    "timemap": "http://www.pastpages.org/timemap/link/",
    "timegate": "http://www.pastpages.org/timegate/"
  },
  {
    "id": "ba",
    "name": "Bibliotheca Alexandrina Web Archive",
    "timemap": "http://web.archive.bibalex.org/web/timemap/link/",
    "timegate": "http://web.archive.bibalex.org/web/"
  },
  {
    "id": "blarchive",
    "name": "UK Web Archive",
  "timemap": "http://www.webarchive.org.uk/wayback/archive/timemap/link/",
  "timegate": "http://www.webarchive.org.uk/wayback/archive/",
},
{
  "id": "loc",
  "name": "Library of Congress",
  "timemap": "http://webarchive.loc.gov/all/timemap/link/",
  "timegate": "http://webarchive.loc.gov/all/",
},
{
  "id": "archiveit",
  "name": "Archive-It",
  "timemap": "http://wayback.archive-it.org/all/timemap/link/",
  "timegate": "http://wayback.archive-it.org/all/",
},
{
  "id": "ukparliament",
  "name": "UK Parliament Web Archive",
  "timemap": "http://webarchive.parliament.uk/timemap/",
  "timegate": "http://webarchive.parliament.uk/timegate/"
},
{
  "id": "uknationalarchives",
  "name": "UK National Archives Web Archive",
  "timemap": "http://webarchive.nationalarchives.gov.uk/timemap/",
  "timegate": "http://webarchive.nationalarchives.gov.uk/timegate/",
},
{
  "id": "archive.is",
  "name": "archive.today",
  "timemap": "http://archive.today/timemap/",
  "timegate": "http://archive.today/timegate/",
},
{
  "id": "is",
  "name": "Icelandic Web Archive",
  "timemap": "http://wayback.vefsafn.is/wayback/timemap/link/",
  "timegate": "http://wayback.vefsafn.is/wayback/",
},
{
  "id": "swa",
    "name": "Stanford Web Archive",
    "timemap": "https://swap.stanford.edu/timemap/link/",
    "timegate": "https://swap.stanford.edu/",
  },
  {
    "id": "pt",
    "name": "Portuguese Web Archive",
    "timemap": "http://arquivo.pt/wayback/timemap/*/",
    "timegate": "http://arquivo.pt/wayback/",
  },
  {
    "id": "perma",
    "name": "Perma Archive",
    "timemap": "https://perma-archives.org/warc/timemap/*/",
    "timegate": "https://perma-archives.org/warc/timegate/",
  },
  {
    "id": "nrscotland",
    "name": "National Records of Scotland",
    "timemap": "http://webarchive.nrscotland.gov.uk/timemap/",
    "timegate": "http://webarchive.nrscotland.gov.uk/timegate/"
  }
]
```

APPENDIX C

MEMENTO DAMAGE API OUPUT

Negotiation in other dimensions beyond time (Chapter 6) requires surfacing attributes about mementos. In Section 6.2.2 we described "derived attributes", to which lengthly calculation like the Memento Damage API at `http://memento-damage.cs.odu.edu/api/` would be attributes. The numerical value to be used by the damage score is highlighted in the sample output from the API below.

```
{
  "csses": [],
  "archive_time": "2018-03-26T14:37:37.778962",
  "redirect_uris": [
    [
      "https://web.archive.org/web/20020525205220/http://clarkjolley.com/",
      200,
      "text/html; charset=utf-8"
    ]
  ],
  "iframes": [],
  "potential_damage": {
    "text": 0.022399999999999996,
    "image": 0.17374950983349416,
    "multimedia": 0,
    "js": 0.05,
    "iframe": 0,
    "total": 0.24614950983349418,
    "css": 0
  },
  "weight": {
    "text": 0.09999999999999998,
    "image": 0.29999999999999993,
    "multimedia": 0.5,
    "js": 0.05,
    "iframe": 0.29999999999999993,
    "css": 0.05
  },
  "jses": [
    {
      "url": "https://archive.org/includes/analytics.js?v=cf34f82",
      "redirect_urls": [
        [
          "https://archive.org/includes/analytics.js?v=cf34f82",
          200,
          "application/x-javascript"
        ]
      ],
      "content_type": "application/x-javascript",
      "potential_damage": {
        "total": 1
      },
      "status_code": 200
    }
  ],
  "total_damage": 0.34072227221225737
  ],
  "message": "The damage calculation took 17 seconds",
  "is_archive": true
}
```

APPENDIX D

SAMPLE WEB SERVICE FOR CONTENT-BASED ATTRIBUTE

A sample Web service that takes a URI and returns the status code.

```
package main

import (
        "fmt"
        "github.com/gorilla/mux"
        "log"
        "net/http"
        "time"
)

func main() {
        r := mux.NewRouter().SkipClean(true)
        r.PathPrefix("/").HandlerFunc(
                func(w http.ResponseWriter, r *http.Request) {
                        log.Println(r.RequestURI)
                        urim := r.RequestURI[1:] // Parse out URI-M
                        var netClient = &http.Client{
                                Timeout: time.Second * 10,
                                CheckRedirect: func(req *http.Request, via
                                ↪ []*http.Request) error {
                                        return http.ErrUseLastResponse
                                },
                        }
                        response, _ := netClient.Get(urim)
                        fmt.Fprint(w, response.StatusCode)
        })

        http.ListenAndServe(":1209", r)
}
```

APPENDIX E

PWAA REFERENCE IMPLEMENTATION

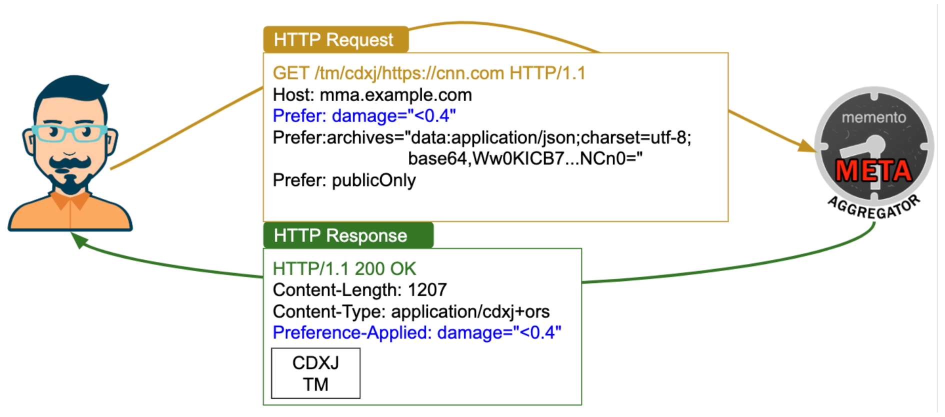
This appendix provided a reference implementation to exhibit the Private Web Archive Adapter (pwaa) mementity in the Mementity Framework. The latest code is available at https://github.com/machawk/pwaa.

```
package main

import (
        "encoding/json"
        "fmt"
        "github.com/google/uuid"
        "gopkg.in/oauth2.v3/models"
        "log"
        "net/http"

        "gopkg.in/oauth2.v3/errors"
        "gopkg.in/oauth2.v3/manage"
        "gopkg.in/oauth2.v3/server"
        "gopkg.in/oauth2.v3/store"
)

func main() {
        // Setup token storage
        manager := manage.NewDefaultManager()
        manager.SetAuthorizeCodeTokenCfg(manage.DefaultAuthorizeCodeTokenCfg)
        manager.MustTokenStorage(store.NewMemoryTokenStore())
        clientStore := store.NewClientStore()
    manager.MapClientStorage(clientStore)

        srv := server.NewDefaultServer(manager)
        srv.SetAllowGetAccessRequest(true)
        srv.SetClientInfoHandler(server.ClientFormHandler)
        manager.SetRefreshTokenCfg(manage.DefaultRefreshTokenCfg)

        srv.SetInternalErrorHandler(func(err error) (re *errors.Response) {
                log.Println("Internal Error:", err.Error())
                return
        })

        srv.SetResponseErrorHandler(func(re *errors.Response) {
                log.Println("Response Error:", re.Error.Error())
        })

        http.HandleFunc("/token", func(w http.ResponseWriter, r
        ↪ *http.Request) {
                srv.HandleTokenRequest(w, r)
        })

        http.HandleFunc("/credentials", func(w http.ResponseWriter, r
        ↪ *http.Request) {
                clientId := uuid.New().String()[:8]
                clientSecret := uuid.New().String()[:8]
                err := clientStore.Set(clientId, &models.Client{
                        ID:     clientId,
                        Secret: clientSecret,
                        Domain: "http://localhost:1210",
                })
                if err != nil {
                        fmt.Println(err.Error())
                }

                w.Header().Set("Content-Type", "application/json")
                json.NewEncoder(w).Encode(map[string]string{"CLIENT_ID":
                ↪ clientId, "CLIENT_SECRET": clientSecret})
        })

        http.HandleFunc("/protected", validateToken(func(w
        ↪ http.ResponseWriter, r *http.Request) {
                w.Write([]byte("Hello, I'm protected"))
        }, srv))

        http.ListenAndServe(":1210", nil)
}

func validateToken(f http.HandlerFunc, srv *server.Server) http.HandlerFunc {
        return http.HandlerFunc(func(w http.ResponseWriter, r *http.Request)
        ↪ {
                _, err := srv.ValidationBearerToken(r)
                if err != nil {
                        http.Error(w, err.Error(), http.StatusBadRequest)
                        return
                }

                f.ServeHTTP(w, r)
        })
}
```

APPENDIX F

STARGATE INTERACTION WALKTHROUGH

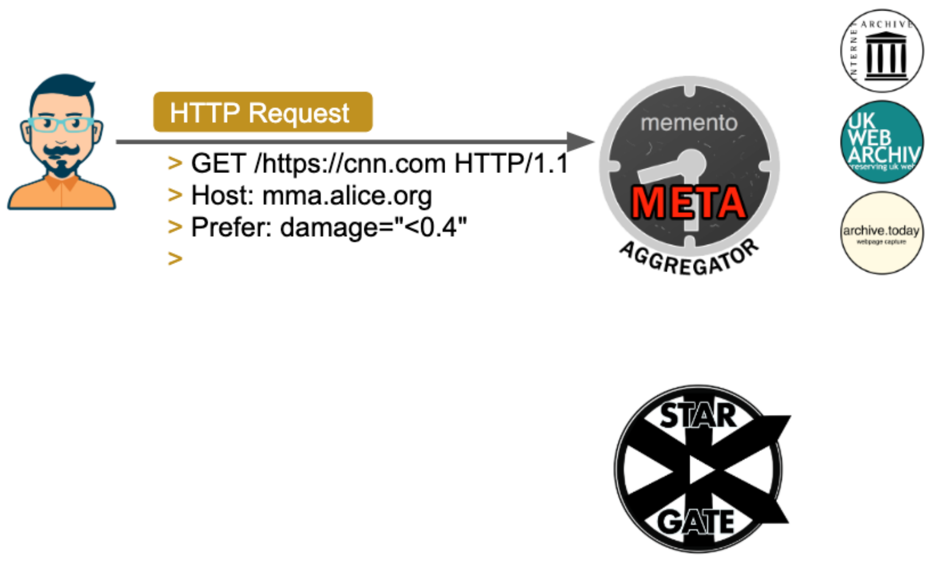
Section 7.2.1 describes a user interacting with a StarGate through a Memento Meta-Aggregator. In this appendix we provide an example of a walk-through of interacting with an MMA and StarGate using HTTP Prefer.



```
GET /tm/cdxj/https://cnn.com HTTP/1.1
Host: mma.example.com
Prefer: damage="<0.4"
Prefer:archives="data:application/json;charset=utf-8;
                 base64,Ww0KICB7...NCn0="
Prefer: publicOnly
```

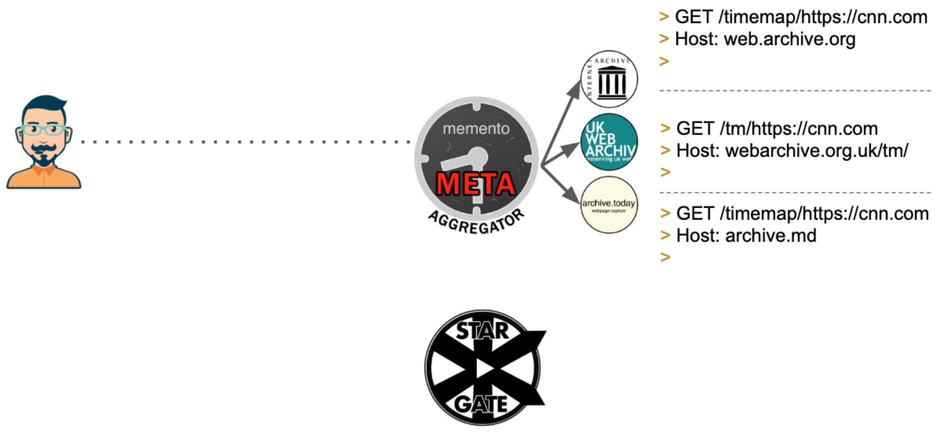
```
HTTP/1.1 200 OK
Content-Length: 1207
Content-Type: application/cdxj+ors
Preference-Applied: damage="<0.4"
```

Fig. 96 A client sends a request to an MMA with HTTP Prefer headers. The MMA responds with only a single preference being applied. The other figures in this appendix walkthrough the dynamics in more detail.



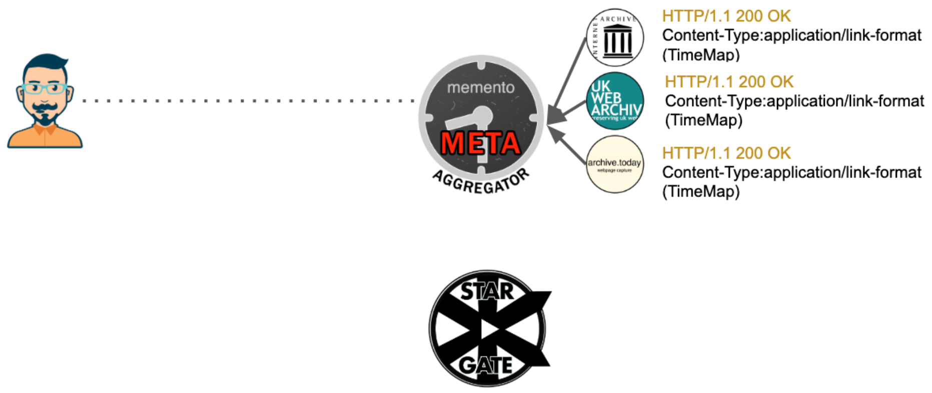
```
> GET /https://cnn.com HTTP/1.1
> Host: mma.alice.org
> Prefer: damage="<0.4"
>
```

Fig. 97 Bob sends a request to an MMA, specifying that he want a TimeMap to only contain mementos below a damage threshold.



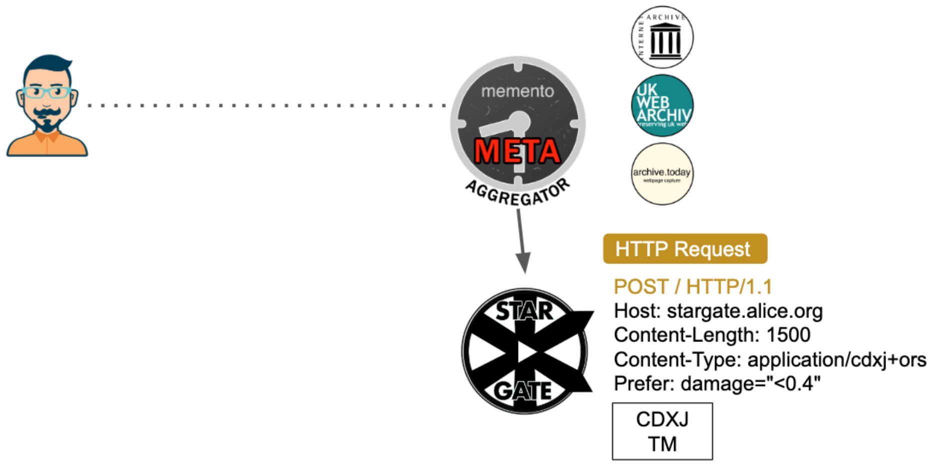
```
> GET /timemap/https://cnn.com
> Host: web.archive.org
>

> GET /tm/https://cnn.com
> Host: webarchive.org.uk/tm/
>

> GET /timemap/https://cnn.com
> Host: archive.md
>
```

Fig. 98 The MMA requests captures from CNN from the three archives with which it is configured.



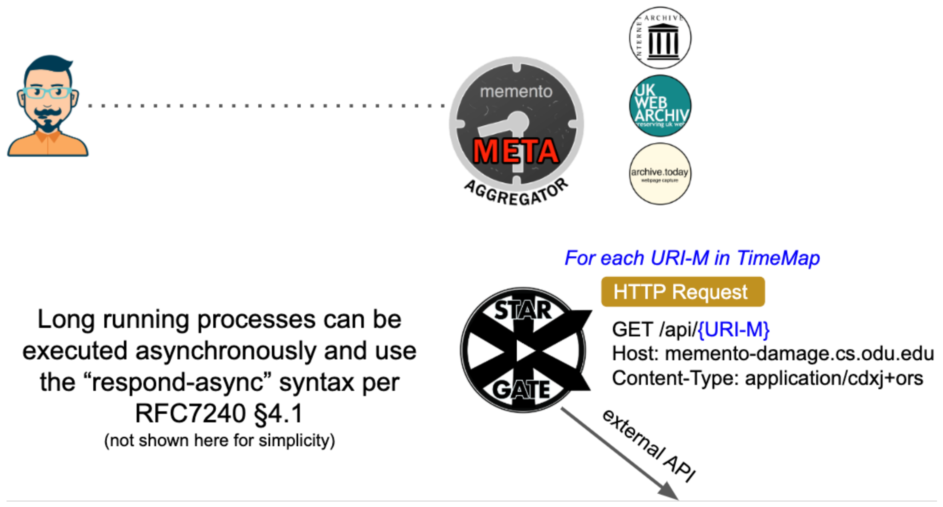
```
HTTP/1.1 200 OK
Content-Type:application/link-format
(TimeMap)

HTTP/1.1 200 OK
Content-Type:application/link-format
(TimeMap)

HTTP/1.1 200 OK
Content-Type:application/link-format
(TimeMap)
```

Fig. 99 The three archives respond with their respective TimeMaps for the URI-R.



```
POST / HTTP/1.1
Host: stargate.alice.org
Content-Length: 1500
Content-Type: application/cdxj+ors
Prefer: damage="<0.4"
```

Fig. 100 The MMA sends the aggregated TimeMap to a StarGate using an HTTP POST and relaying the specified preference.


memento
META
AGGREGATOR
For each URI-M in TimeMap
HTTP Request
STAR
GATE
external API
Long running processes can be executed asynchronously and use the "respond-async" syntax per RFC7240 §4.1
(not shown here for simplicity)

```
GET /api/{URI-M}
Host: memento-damage.cs.odu.edu
Content-Type: application/cdxj+ors
```

Fig. 101 The StarGate queries a Web service to obtain a value for each URI-M. This procedure may take substantial time per Section 7.2.1.

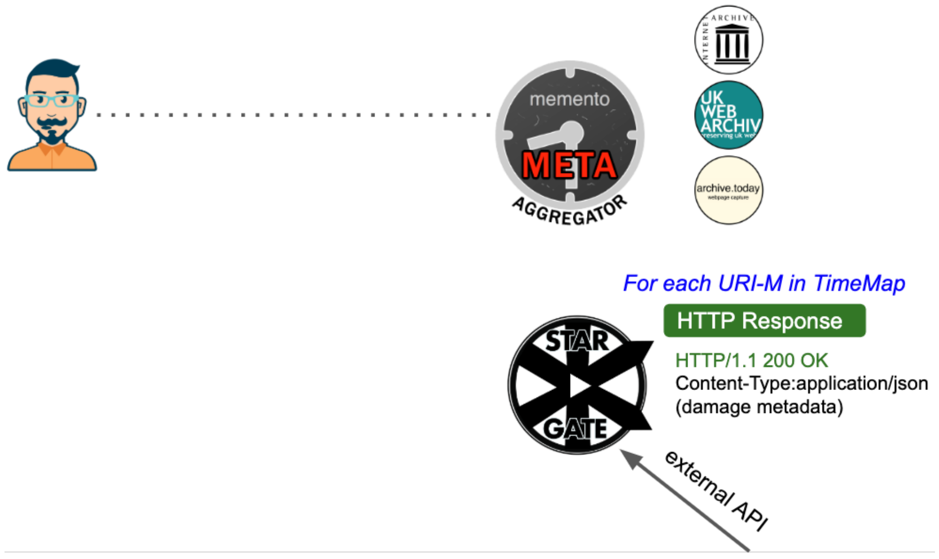


Fig. 102 The external Web service returns a value for each URI-M.

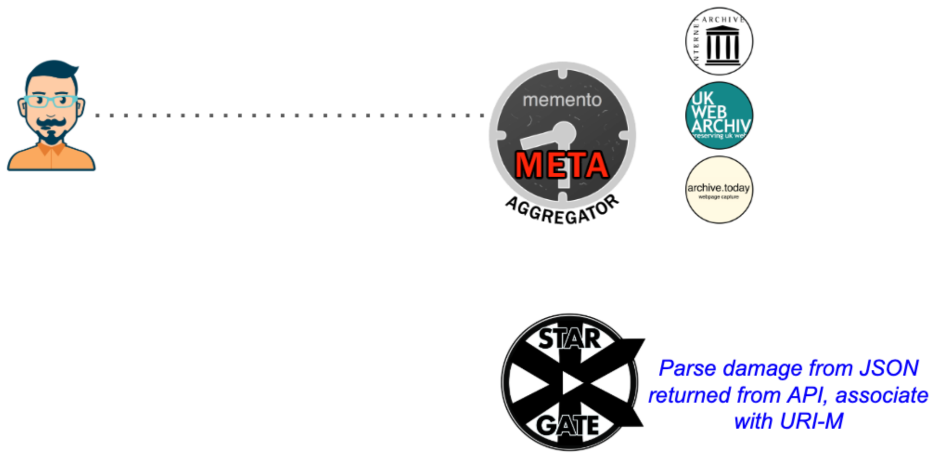

memento
META
AGGREGATOR
STAR
GATE
Parse damage from JSON returned from API, associate with URI-M
archive.today

Fig. 103 The StarGate extracts the damage values from the HTTP responses from the Web service.

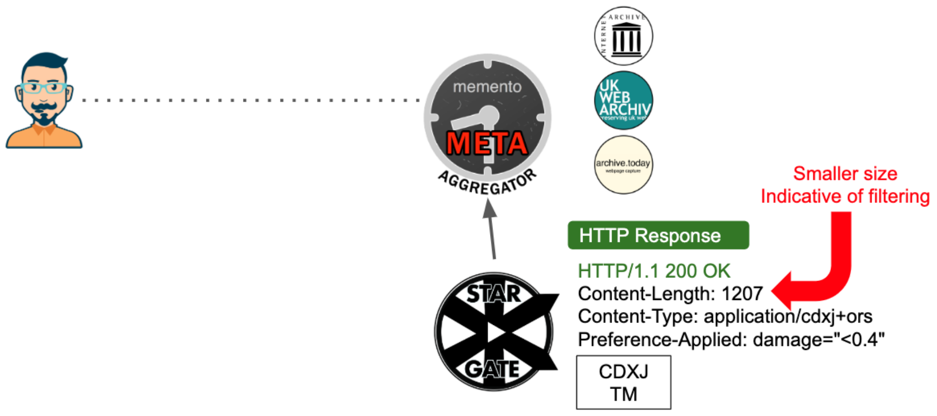


Fig. 104 The StarGate associates each value for the respective URI-M and filters the StarMap to only contain references to mementos that are within the specified preference threshold.

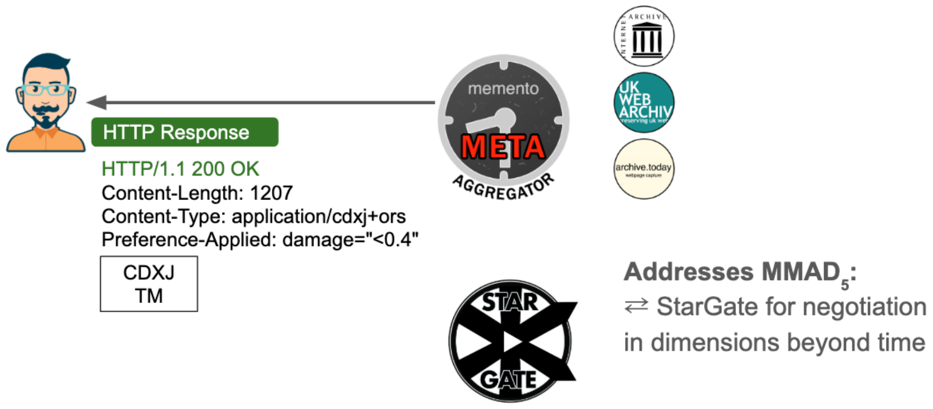


Fig. 105 The MMA returns the StarMap response to the client with an indication that the preference was applied using the `Preference-Applied` HTTP header.

```
curl -v -H 'Prefer: damage="<0.4"'
        -H 'Prefer: archives="data:application/json;charset=utf-8;base64,Ww0KICB7...NCn0="'
        -H 'Prefer: publicOnly' https://mma.example.com/https://cnn.com

> GET /tm/cdxj/https://cnn.com HTTP/1.1
> Host: mma.example.com
> User-Agent: curl/7.54.0
> Accept: */*
> Prefer: damage="<0.4"
> Prefer: archives="data:application/json;charset=utf-8;base64,Ww0KICB7...NCn0="
> Prefer: publicOnly
>
< HTTP/1.1 200 OK
< Server: nginx
< Date: Fri, 19 April 2019 13:42:55 GMT
< Content-Type: application/cdxj+ors
< Vary: *
< Front-End-Https: on
< Preference-Applied: damage="<0.4"
<
!context ["http://tools.ietf.org/html/rfc7089"]
!id {"uri": "https://mma.example.com/tm/cdxj/https://cnn.com"}
!keys ["memento_datetime_YYYYMMDDhhmmss"]
!meta {"original_uri": "https://cnn.com"}
!meta {"timegate_uri": "https://mma.example.com/tg/https://cnn.com"}
!meta {"timemap_uri": {"link_format": "https://mma.example.com/tm/link/http://cnn.com",
↪ "json_format": "https://mma.example.com/tm/json/http://cnn.com", "cdxj_format":
↪ "https://mma.example.com/tm/cdxj/http://cnn.com"}}
20000620180259 {"uri": "https://arquivo.pt/wayback/20000620180259/http://cnn.com/", "rel":
↪ "first memento", "datetime": "Tue, 20 Jun 2000 18:02:59 GMT", "damage": "0.256"}
20000620180259 {"uri": "https://arquivo.pt/wayback/20000620180259/http://cnn.com/", "rel":
↪ "memento", "datetime": "Tue, 20 Jun 2000 18:02:59 GMT", "damage": "0.389"}
20000620180259 {"uri": "http://wayback.vefsafn.is/wayback/20000620180259/http://cnn.com/",
↪ "rel": "memento", "datetime": "Tue, 20 Jun 2000 18:02:59 GMT"}
20000620180259 {"uri": "http://web.archive.org/web/20000620180259/http://cnn.com:80/", "rel":
↪ "memento", "datetime": "Tue, 20 Jun 2000 18:02:59 GMT", "damage": "0.0"}
...
```

Fig. 106 The MMA interaction illustrated in the preceding images in this appendix can be accomplished using curl. Shown here is an abbreviated interaction using the procedure and curl to obtain URI-Ms for a URI-R that are under a damage threshold. Note that despite three preferences being requested, only one is indicated as having been applied by the MMA and StarGate.

VITA

Matthew R. Kelly
Department of Computer Science
Old Dominion University
Norfolk, VA 23529
e-mail: me@matkelly.com

Education

Doctor of Philosophy in Computer Science (2019)
Old Dominion University, Norfolk, Virginia USA
Dissertation: *Aggregating Private and Public Web Archives Using the Mementity Framework*

Master of Science in Computer Science (2012)
Old Dominion University, Norfolk, Virginia USA
Thesis: *An Extensible Framework for Creating Personal Archives of Web Resources Requiring Authentication*

Bachelor of Science in Computer Science (2006)
University of Florida, Gainesville, Florida USA

Publications

An updated list of publications is available at https://matkelly.com/pubs.

Further Information and Updates

A more comprehensive and current version of this vita is accessible at
https://matkelly.com/cv.

Typeset using LaTeX.