Final Report - Dodging the Memory Hole

Mat Kelly
Department of Computer Science
Old Dominion University, Norfolk, VA 23529
mkelly@cs.odu.edu

Institutional Web archives are often responsible for selecting the part of the live Web that is preserved. Individual (personal) Web archivists take it upon themselves to preserve what they feel is important using available tools. These personal Web archives, however, are often inaccessible to others or siloed to the archivists' own machine for privacy, safety (to ensure contents are not deleted), and control of the content. Personal Web archives are often given less credence or authenticity of an accurate account of the Web how it was. Further, personal captures will often contain personalized content from the perspective of the archivist, potentially with personally identifiable information or private information that the archivist may not be initially aware is being sharing.

Archiving online news is the particular focus of this work, as personal Web archivists rightly consider what they see in their browser for online news Web sources as culturally significant. Delegation of the act of preservation through submission of a URI to an institutional Web archive may not prove sufficient in terms of the resulting archival quality and requires the capture to reside elsewhere. Because content on online news Web sites changes rapidly as more stories are reported, it is important to enable individuals to capture what they see at that moment.

In attending the Dodging the Memory Hole (DtMH) conference, I became informed of the impending technical needs of personal Web archivists attempting to preserve an accurate account of the historical record through archiving online news. The purpose of my work for DtMH was to investigate and implement tools that allow personal Web archivists to accomplish three goals in anticipation of some needs I extrapolated from conference attendees:

1. **Aggregate** references to their archival holding with those from archival institutions.
2. Securely **propagate** the content of their archives for longevity.
3. Systematically **regulate access** of their captures.

These goals required extending the functionality of an archival aggregator, implementing secure propagation to a previously developed archival dissemination tool, and creating a new tool in the archival hierarchy to regulate access, respectively. Details for the efforts applied for each of these goals are described in the following sections.

1 Aggregate

I had previously been involved in the implementation of an open source Memento aggregator [2]. Memento is a recognized standard protocol that applies the dimension of time to the Web through content negotiation and association of captures of content [8]. The previously developed Memento aggregator, MemGator, allows anyone with a list of archives to easily run an aggregator to return results only from that set of archives. MemGator is

written in the Go programming language[1], with which I was not technically familiar prior to this task.

My objective for DtMH was to extend MemGator to allow clients to request additional archives to be queried at runtime (i.e., "just-in-time" instead of manual pre-configuration) to be included in the aggregation. The product of aggregation in Memento is a TimeMap, which consists of a standardly formatted list of URIs and select respective metadata as well as identifiers for other entities like the original resource (the live Web URI).

MemGator functions in one of two different modes:

- One-off Mode - user requests TimeMap for a URI on the command line.
- Server Mode - user deploys persistent service to handle requests for TimeMap by other users using HTTP.

The set of archives MemGator queries by default is specified in an online JSON file. A user may also specify this set at runtime for the One-off Mode or to be used with each query (i.e., the set of archives is static and unchangeable by the client) upon request to the service in Server Mode. In this work, I focus on the Server Mode of MemGator.

My implementation sought to allow anonymous users to supply additional archives to be queried for captures. This is particularly relevant to personal Web archives being included in a TimeMap. For example, if I preserve the state of my local online newspaper that all other institutional Web archives seemed to have missed (Figure 1, callout A) and the page on the live Web has since changed, I can see my capture (callout B) listed temporally inline with the institutions'.

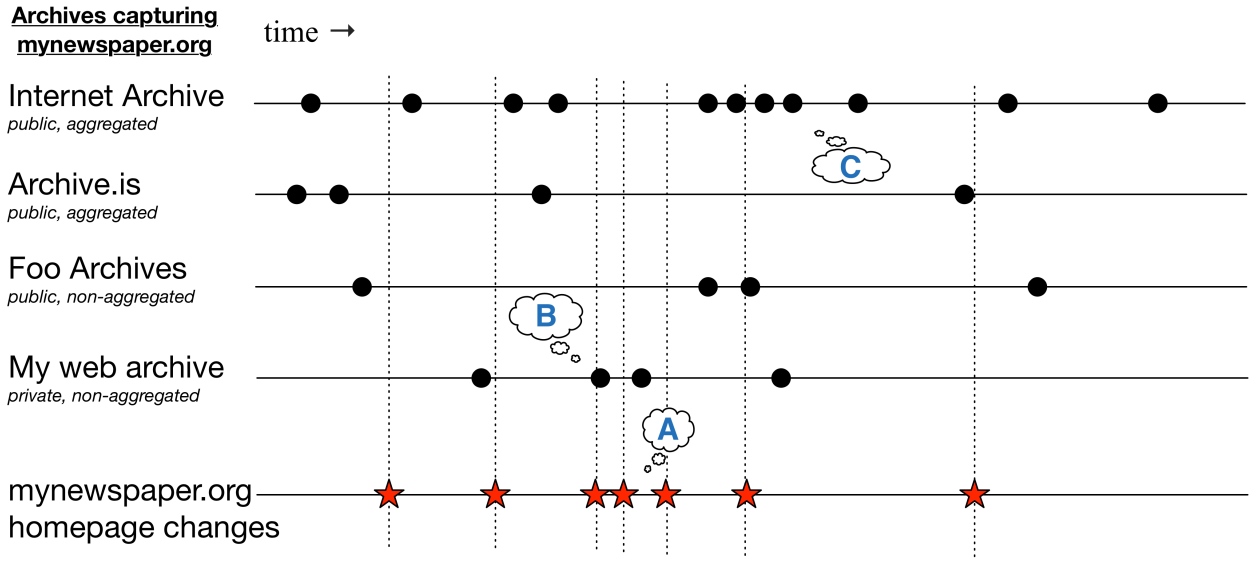

Figure 1: Captures of a local newspaper that interests a user and thus is better preserved if all captures are aggregated.

[1] `https://golang.org/`

In the Server Mode, a user may query a MemGator instance for a TimeMap for a specified URI-R. Figure 2 shows the typical operation of a MemGator instance in Server Mode. Upon starting, the instance fetches the metadata for the known set of Web archives then listens on the network for a request from a client. The parameters from a client come embedded in the URI requested. For instance, if a user that wants to obtain a TimeMap for The Maneater would request the URI:
`http://memgatorhost/timemap/json/http://www.themaneater.com`

...where each portion of the URI is representative of one facet of the request:

- *memgatorhost* is representative of where the MemGator instance is deployed.
- *timemap* specifies that a user wants a Memento TimeMap returned.
- *json* is the format of the TimeMap with alternative of the more standard "link" format [6] or the up-and-coming "cdxj" [3] format.
- *http://www.themaneater.com* is the URI-R being requests, appended to the end of the URI to give the other parameters in the request context.

Memento aggregators inclusive of MemGator[2] are inherently limited in the parameters that can be specified by a user within a URI. I investigated other approaches and mechanisms that would allow a user to specify an additional list of Web archives for aggregation to supplement the results already aggregated by MemGator. Where the URI-R contained within the request (per above) constitutes a "sub-URI" for explanation's sake, the URI for the request to MemGator would become ambiguous and less semantic were multiple sub-URIs contained within. For example:

```
http://memgatorhost/timemap/json/http://myadditionalarchive
  /timemap/http://www.themaneater.com
```

...demonstrates the potential confusion of specifying the URI-R and the additional archive. This ambiguity would be exacerbated were multiple sub-URIs specified in the MemGator URI.

In lieu of specifying additional archives in the URI, I looked into using HTTP request headers for specification. The Web Linking [6] standard serves as the basis for Memento where the semantic relevance (e.g., first memento, TimeMap, etc) can be explicitly specified. For example, within the HTTP response headers for a URI-M, a Link header will specify `<http://www.themaneater.com/>; rel="original"` indicating the live Web URI. This format is useful but syntacticly verbose and further, is only defined for HTTP responses, not HTTP requests, as needed by the requesting user.

Figure 2 graphically displays the user's involvement in requesting a URI-R from an aggregator. No mechanism exists to specify additional archives. After the initial investigation, I modified the MemGator code to, at time of request, check the request from the user for

[2]For example, the non-MemGator Memento aggregator instance at `http://timetravel.mementoweb.org/timemap/link/http://www.themaneater.com`

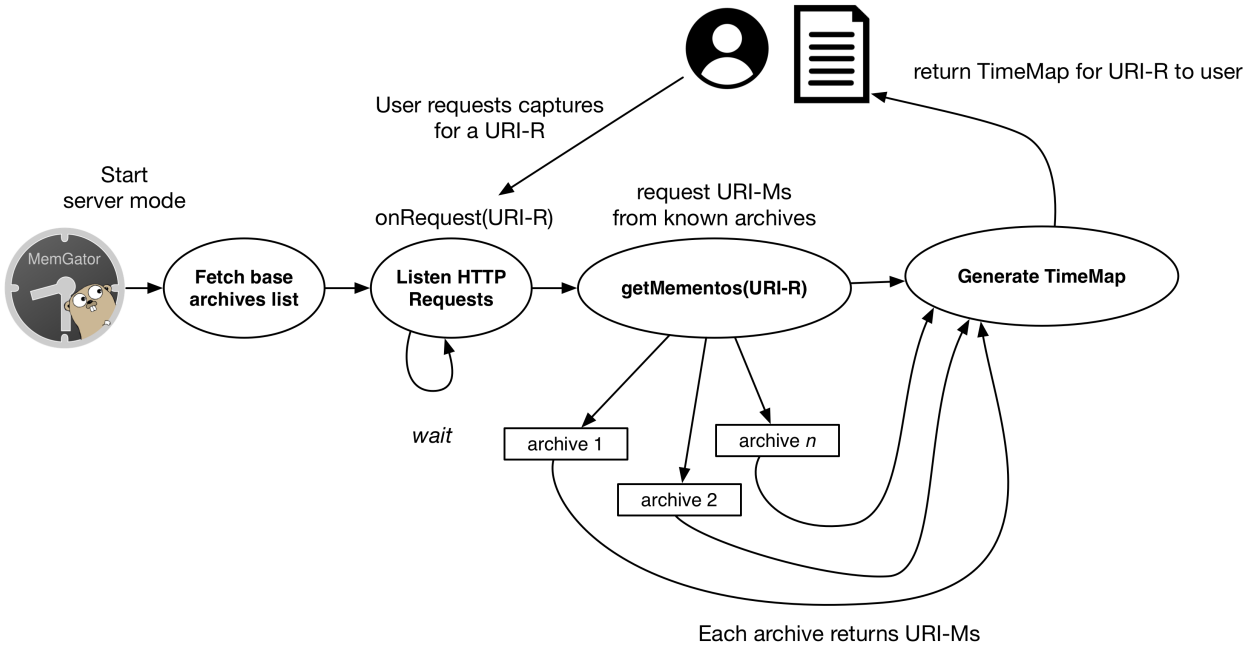

Figure 2: MemGator server mode

an additional HTTP request header, which I named, "X-More-Archives". The "X-" prefix is a common method for specifying non-standard headers. Through this header, a user can specify a delimited list of TimeMap endpoints. While this is easily accomplished through command-line tools like "curl", the idea is that tools may build on this newfound ability and specify this header to MemGator behind the scenes based on values supplied by the user. Figure 4 demonstrates where in the MemGator flow diagram (Figure 2) MemGator must intercept the request from the client, prior to querying the archives. With the assumption that an archive is Memento-compatible, MemGator will treat it equally to the others queried.

As a proof-of-concept, I created a Web archive (WARC) file of `http://www.themaneater.com` using WARCreate [5] – a tool I had previously developed to enable users to create web archives from Google Chrome on their own machine. I then created a collection (named "dtmh") and added the WARC to the collection using a production archival replay system, pywb[3]. I then curl'd the modified MemGator instance without the additional headers via:

```
curl "http://memgatorHost/timemap/json/http://www.themaneater.com"
```

Among the results returned, the "last" memento JSON block was the following:

```
"last": {
  "datetime": "2016-11-28T01:14:00Z",
  "uri": "http://web.archive.org/web/20161128011400/http://www.themaneater.com/"
}
```

[3] `https://github.com/ikreymer/pywb`

I then performed the same HTTP via curl (line breaks added for formatting) with the following command:

```
curl -H "X-More-Archives: http://myLocalWebArchive/dtmh/timemap/*/"
  "http://memgatorHost/timemap/json/http://www.themaneater.com"
```

The "last" memento returned was then:

```
"last": {
  "datetime": "2016-12-01T22:24:42Z",
  "uri": "http://myLocalWebArchive/dtmh/20161201222442mp_/http://www.themaneater.com/"
}
```

Viewing this URI in the browser displays the capture (Figure 3).

With the intention of MemGator being accessed by multiple users, the specification of additional archives is done on a per-request basis; that is, the header must be specified with each request. Future work would involve creating a hash-based id representative of the superset of archives created with the addition of the additional user-specified archives.

2 Encryption and Dissemination

Institutional Web archives are at an advantage compared to personal Web archives in that they are much more resilient to errors. Individual archivists are normally at the mercy of a more limited set of technical hardware. An on-site power surge, for example, is much more likely to be detrimental to a personal than institutional Web archive. In previous work I investigated applying the distributed peer-to-peer InterPlanetary File System (ipfs) to Web archives (WARCs) and created a prototype, InterPlanetary Wayback (ipwb) [1, 4] to accomplish this. In this work, I sought to extend ipwb to mitigate the requirement for personal Web archivists to keep their captures on-site.

Because of the nature of personal Web archival content, sensitive data may be contained within. The WARC files representative of the archive payload contain personally identifiable information or, in the case of online news, an indication of interest by an archivists that might be leveraged elsewhere by others. Regardless of the potential ramifications of disseminating personal Web archives, my goal was to allow personal web archivists to use ipwb to disseminate their content into ipfs using encryption to ensure that they have another means of regulating access and obfuscating what they have preserved.

Once implemented, I reused the WARC file from Section 1. ipwb works by first indexing a WARC using the command

```
ipwb index (warc-path)
```

...which creates a CDXJ file (Figure 5, red circles). Performing this command on The Maneater WARC produces the following record (split into multiple lines here but normally on a single line) for the home page in addition to a CDXJ record for each embedded resources (e.g., images, not shown here) contained in the WARC:

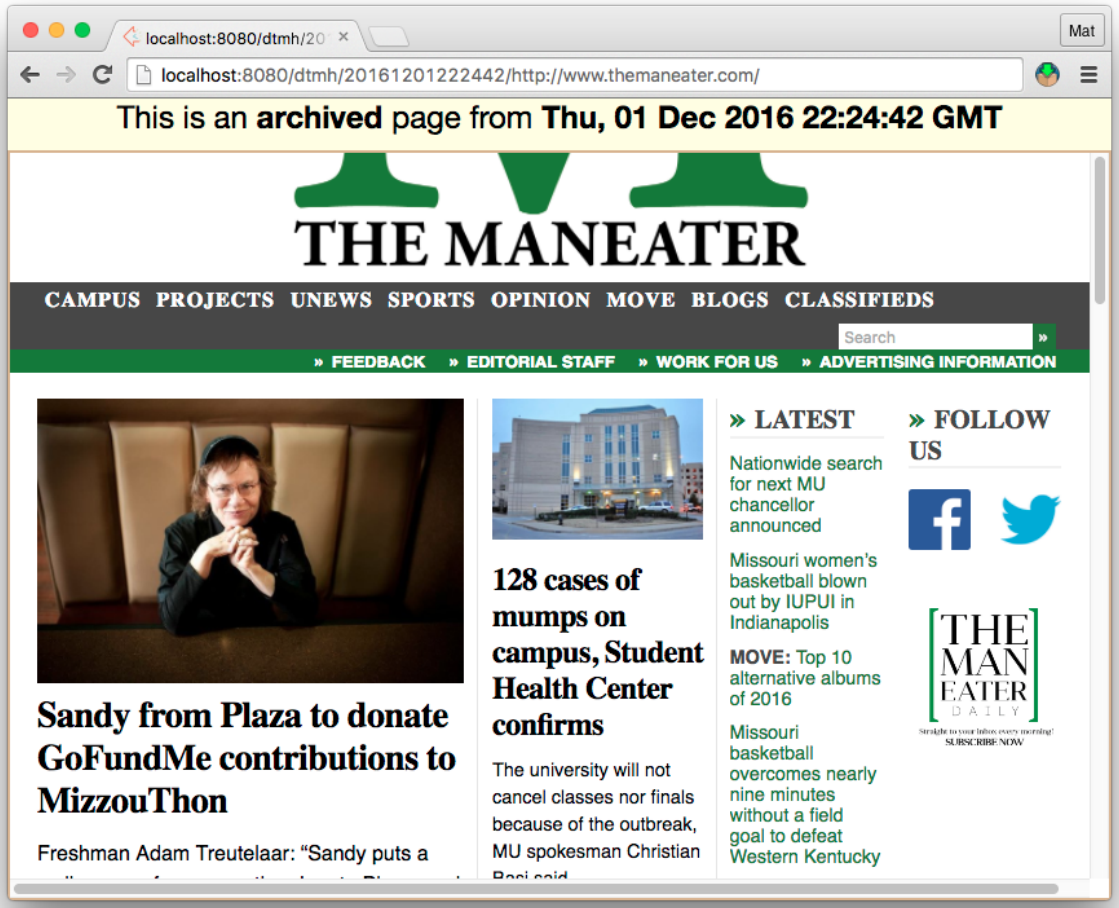

Figure 3: The Maneater homepage, with the URI-M that was included in an aggregated TimeMap from the MemGator modified in this work.

```
com,themaneater)/ 20161201213641 {
  "locator": "urn:ipfs/QmRGooi1C1WpeEXWqvAvooSPmebAbmnNRepUfojDR6KC1S/
              QmWFSGKaQXhTr5yVmXLvGEoysEaPSKR1XqwDbgeXK2YkR6",
  "mime_type": "text/html",
  "status_code": "200"
}
```

The first field is representative of the URI transformed using the less ambiguous SURT format [7], which is a common representation in archival indexes. The second field represents the capture time, December 1, 2016. The mime-type and status code are preserved in the index from the WARC and are necessary for replay. The locator field contains two slash-delimited unique hashes as well as the "urn:ipfs/" prefix to indicate scheme. Both hashes are based on the contents of the HTTP header and entity body, respectively, and are unique

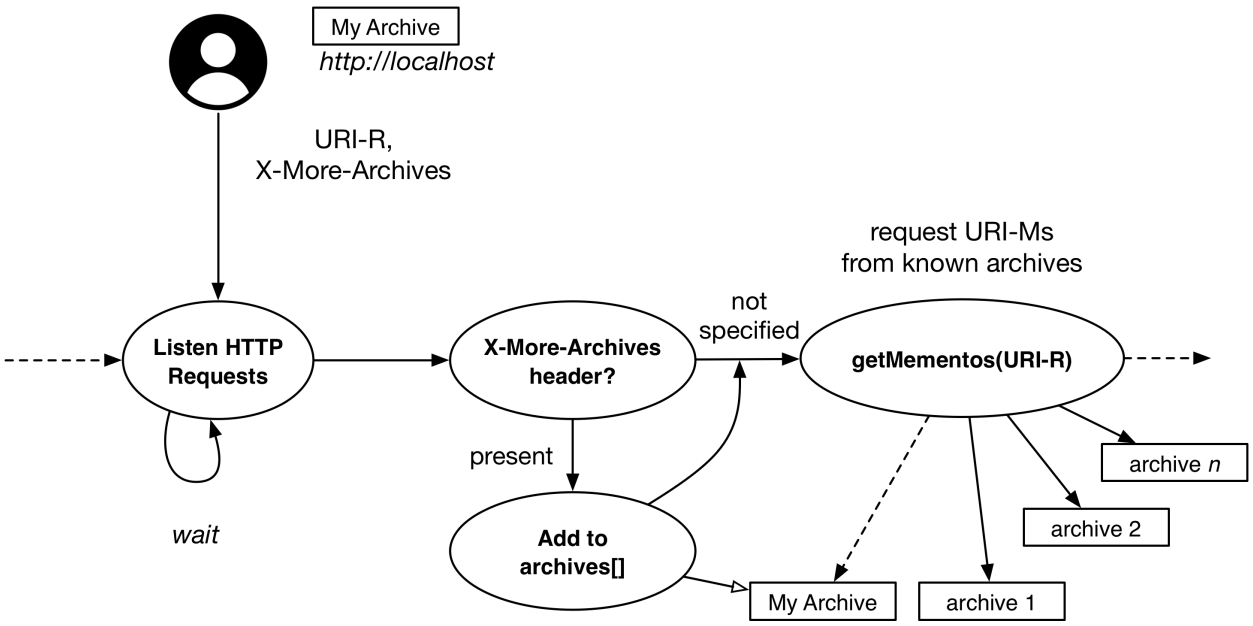

Figure 4: User specification of additional archives

to the content; i.e., if the content changes, these hashes, too, will change. As a simple proof, using the command

```
ipfs cat QmWFSGKaQXhTr5yVmXLvGEoysEaPSKR1XqwDbgeXK2YkR6
```

returns the content beginning with:

```
<!DOCTYPE html><html class="js"><head><title>The Maneater</title>
```

In this work, I introduced an additional flag into the ipwb indexer script as well as implemented the appropriate handling of the additional functionality in the indexing and replay system. Using the same WARC, one can now run the command:

```
ipwb index -e (warc-path)
```

Upon invoking this command, ipwb asks the user for a key. This key is used as a naive basis for encryption with more secure methods to be investigated in the future. Upon entering the key, "dtmh", following CDXJ line is produced for the same content:

```
com,themaneater)/ 20161201221359 {
  "locator": "urn:ipfs/QmVJUGTvUtp26WDwbuFnV4Np6VrGEnu8kPW5TdfjyWMfiN/
              QmXxmHkQPXx1QHkVoNudm6sYdaJv5Esti9ZJmmys5kcxGg",
  "encryption_method": "xor",
  "encryption_key": "dtmh",
  "mime_type": "text/html",
  "status_code": "200"
}
```

Manually running "ipfs cat QmXxmHkQPXx1QHkVoNudm6sYdaJv5Esti9ZJmmys5kcxGg" as before returns a very long string starting with "WFUpJycgNDghVAUcCRhTVAwAAARE-FwEJFwdQSg4" instead of HTML. This string essentially amounts to gibberish, i.e., the encrypted and base64-encoded content. Using ipwb's replay system (Figure 5, blue circles) with this index file, however, allows for decryption and decoding of both the entity body and the HTTP headers.

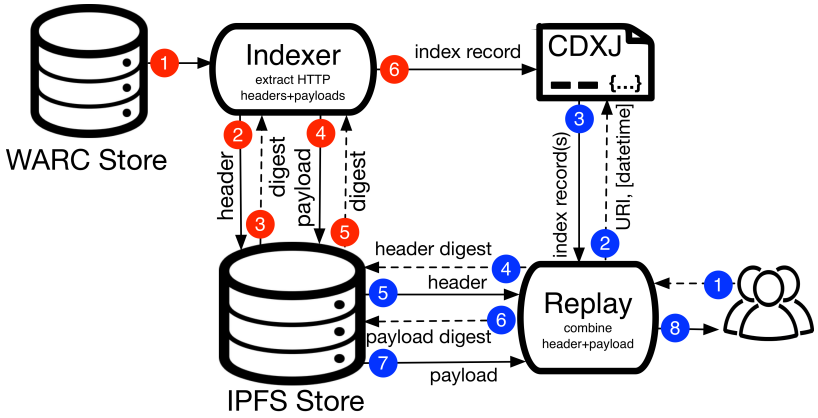

Figure 5: InterPlanetary Wayback indexing and dissemination flow.

Encryption at time of dissemination ensures that private or personal archived Web content is not comprised when transmitted. I utilized a simple XOR block cypher as a proof-of-concept as applied to ipwb. Obfuscating WARC content in this way requires indexing, dissemination, and replay process to all use the same mechanism. In implementing encryption in ipwb, I introduced two additional fields into the generated CDXJ representative of the URI-M, which is also used for remote recall of content in ipfs. The two additional fields specify the encryption method and key. While the latter is safe to include and necessary for symmetric encryption, the "key" would need to be sanitized from derivative indexes prior to sharing of the CDXJ index file.

3 Access Control

The third component of this research was to introduce a mechanism for access control. While this goal was not satisfactorily completed in-time for this final report, future work to integrate the changes in ipwb and MemGator will require a control mechanism for regulating access to URI-Ms newly included in a TimeMap as returned from the improved MemGator.

4 Potential Use Case

The support for funding from DtMH facilitated both the solicitation of requirements in attending the DtMH conference as well as informing the implementations to support these requirements. What follows is a use case newly addressable with the software efforts produced from the support for my attendance of DtMH.

A journalist for an online news Web site is responsible for posting content from her community. While the local news coverage is sparse, the impact of a controversial long-lived

event that is occurring in her community will soon have national ramifications. She fears that those affected by the event described in the narrative of the stories may leverage their power to get the content preemptively removed. Their intention in doing so could prevent tarnishing their reputation and hindering their agenda. For this reason, maintaining a live online archive of the articles related to the story is insufficient, as the interested party would likely share the same fate of takedown as current stories.

The journalist creates Web archives of her news stories for offline preservation in the original context (captures of her online news articles) in case the scenario of removal is realized. She wishes to share the full story of the controversial news via her Web archive captures but to do so with anyone that is interested beyond a select set of individuals. It is critical that the original Web context be accurately replicated and that the content of the stories not be tampered.

The two products of my DtMH funding would directly address this scenario. The modifications to MemGator allow those looking for potentially removed captures of the progressing stories may access her Web archives for which she has exposed the endpoint via the aggregator. To facilitate the Web archive captures from being removed, my previous work on InterPlanetary Wayback would help disseminate her captures; however, the encryption aspect added to indexing and replay of these captures would, as implemented with my DtMH funding, would encourage secure transmission of these archives. Because of the content-based addressing inherent in IPFS, the content of her captures may not be manipulated without being located at a different address and thus not requested by those interested.

The journalist in this scenario could run:

```
ipwb index -e controversialStory.warc > controversialStoryIPWBIndex.cdxj
```

...producing the archival index file containing:

```
org,mylocalnews)/controversialEvent/story3.html 20191104193336 {
  "locator": "urn:ipfs/QmZCVqPB3M4dit5yJwgQNEZj5qJ7yXAbHndewhq3eeLsVj/
                        QmSzEMo9vG6BHcVfGmNxM7jg4X7ut3Mdc7BR27rBVfX2kz",
  "encryption_method": "xor",
  "encryption_key": "mySecretKey",
  "mime_type": "text/html",
  "status_code": "200"
}
```

Sharing the CDXJ file with optionally removing the `encryption_key` field for added security would allow others to view her captures. In the future, personal Web archives are expected to become more prevalent. With the additions I contributed to MemGator as supported by DtMH , anyone may retrieve and view her and others' captures, aggregated with those from institutional archives, without the need for the personal archivist to configure or host the aggregator. The additional functionality introduced in this work to allow this through the, "X-More-Archives" header is described in Section 1.

5 Products

In this work I modified MemGator to support client-side just-in-time inclusion of personal Web archives to aggregated TimeMaps. Additionally, I modified InterPlanetary Wayback to allow for encryption of personalized Web content to be disseminated into IPFS as well as decryption at time of archival replay. The third facet relating to creating a new software package for access control is in the works but first needs more thorough investigation and thus was not sufficiently completed in time for this report. The modified MemGator and InterPlanetary Wayback are available as open source software at the following URIs:

- `https://github.com/machawk1/memgator`
- `https://github.com/oduwsdl/ipwb`

References

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

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

[3] S. Alam, M. L. Nelson, H. Van de Sompel, L. Balakireva, H. Shankar, and D. S. H. Rosenthal. Web Archive Profiling Through CDX Summarization. In *Proceedings of Theory and Practice of Digital Libraries (TPDL)*, pages 3–14, 2015.

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

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

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

[7] K. Sigursson, M. Stack, and I. Ranitovic. Heritrix User Manual. `http://crawler.archive.org/articles/user_manual/glossary.html#surt`.

[8] H. Van de Sompel, M. Nelson, and R. Sanderson. HTTP Framework for Time-Based Access to Resource States – Memento. IETF RFC 7089, December 2013.