WASAPIfying Private Web Archiving Tools for Persistence and Collaboration

Mat Kelly, PhD
Assistant Professor, Information Science
Drexel University, College of Computing & Informatics (CCI)
Philadelphia, PA

mkelly@drexel.edu
https://matkelly.com
@machawk1

IIPC Web Archiving Conference (WAC)
June 15, 2021

Motivation & Objectives

- Use an established approach of pulling and pushing web archive files to/from other services and tools
- Independent endeavor
  - not affiliated with Internet Archive, Archive-It, Webrecorder, WASAPI project

Goal: Programmatically integrate prior work on WASAPI into existing tools

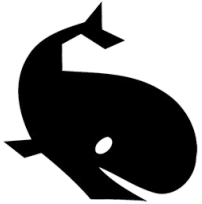
WASAPIfying?

- **WASAPI**: **W**eb **A**rchiving **S**ystems **API**
- IMLS NLG-L grant executed 2016-2017 (#LG-71-15-0174-15)
  - Awarded to Internet Archive, Stanford Libraries, UNT Libraries, Rutgers
  - (Again: No affiliation)
- Implementations created during course of grant
- This work (@ IIPC WAC 2021) is reusing/extending work from the grant into additional tools
  - Goals of furthering interoperability, persistence, and collaboration of data sources.
  - Everything produced here is likewise open source.

Private Web Archive?

- (Private Web) Archive vs. Private (Web Archive)
  - Contents were originally “private” on the web (e.g., behind authentication)
  - The “archive” itself is not publicly accessible (e.g., on a PC)
- Both tools-of-target in this work are intended to create a personal web archive with potential to extend to be:
  - Publicly accessible
  - Aggregated among others’ captures
  - Partially composed of private web content

Personal/Private Web Archiving Tools



Web Archiving Integration Layer (WAIL): https://github.com/machawk1/wail

- Created in 2013, perpetually maintained/evolving
- OpenWayback, Heritrix, others on desktop w/ GUI
- At one time, development funded by National Endowment for Humanities*

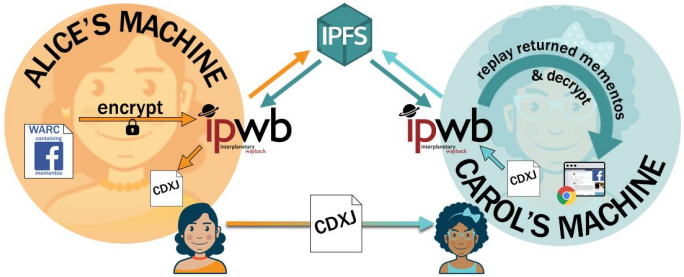
InterPlanetary Wayback (ipwb): https://github.com/oduwsdl/ipwb

- Created at the Archives Unleashed Hackathon 2016
- WARCs + IPFS → distributed personal web archives
- In-collaboration w/ Sawood Alam (now at Internet Archive)

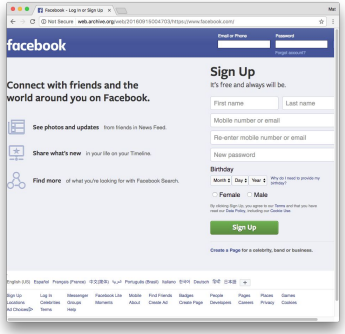
Earlier form of both tools presented at IIPC WAC 2017, London

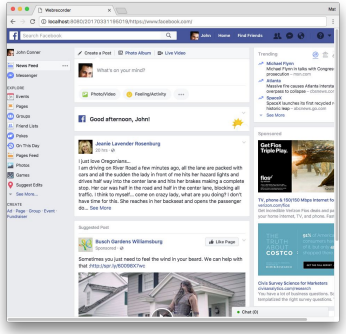
* HD-51670-13
* HK-50181-14

Collaboration



- Collaborative collection building by aggregating WARCs from a variety of sources
- Exchanging captures via WARCs
- Potentially different representations based on personalization, difference in time of capture, etc.





Persistence

- Previously provided by **ipwb**
  - WARC payloads on IPFS
  - Share indexes, others pull contents when Alice goes offline, computer dies
  - Bob can resurface index/captures
- Key contribution in *this* work is to pull from sources for facilitating redundancy onto one's own hardware.

Existing Implementations

Programmatic Libraries

- UNT's py-wasapi-client: https://github.com/unt-libraries/py-wasapi-client
  - Module and CLI executable
- Stanford's Java client: https://github.com/sul-dlss/wasapi-downloader

Clients in the Wild

- Archives Unleashed Cloud, client for WASAPI @ Archive-It

WASAPI Servers

- Archive-It (via Internet Archive)
- Webrecorder.io / Rhizome's Conifer

Existing Implementations

Programmatic Libraries

- UNT's py-wasapi-client: https://github.com/unt-libraries/py-wasapi-client
  - Module and CLI executable
- Stanford's Java client: https://github.com/sul-dlss/wasapi-downloader

Clients in the Wild

- Archives Unleashed Cloud, client for WASAPI @ Archive-It

WASAPI Servers

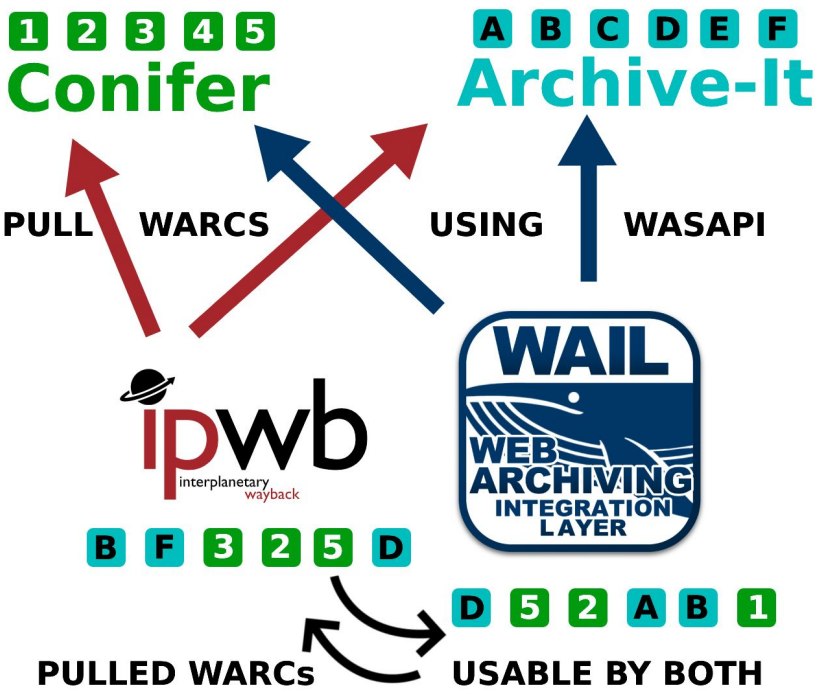
- Archive-It (via Internet Archive)
- Webrecorder.io / Rhizome's Conifer

Timeline

- Original (rough) implementation in 2020 in anticipation of IIPC WAC 2020 (Montréal)
  - Scrapped and re-implemented in Jan 2021
- Progressive and ongoing integration of adaptive layer while ensuring base tool is decoupled for potential reuse.

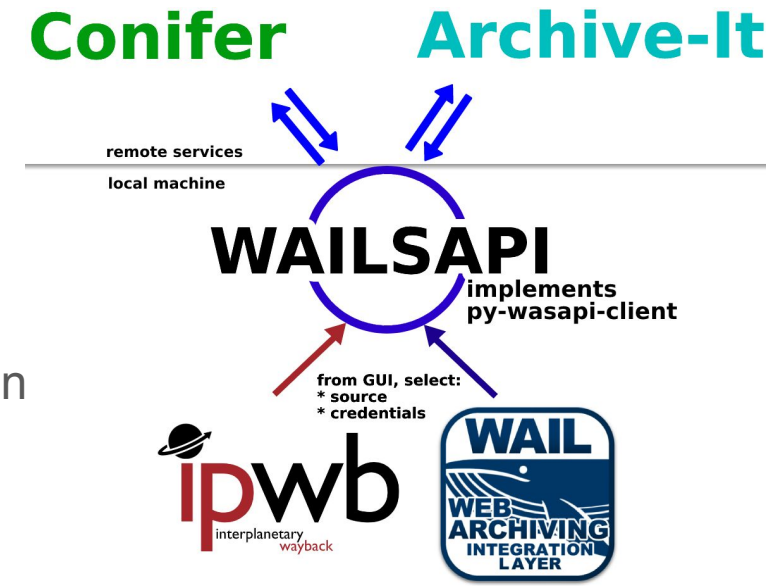
High-level Implementation Goals

- {WAIL, ipwb} pull WARCs from {Archive-It, Conifer}
  - Verify pulled WARCs are replayed
- Rigorous testing not-yet-implemented and out-of-scope of this initial effort.
- This work is a rudimentary, initial effort to determine further feasibility.



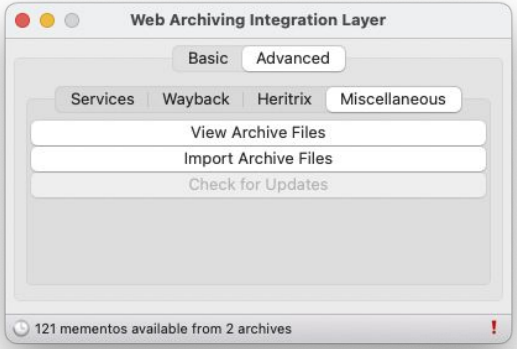
Implementation

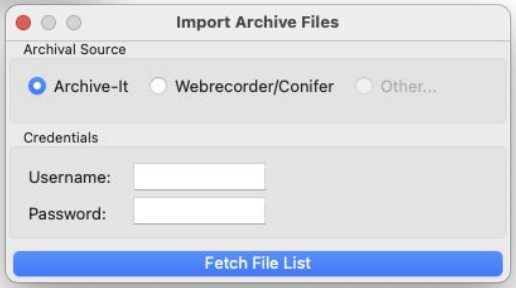
- Create REST endpoint (WAILSAPI) at localhost, bundle with tools
- Provide UI elements to allow user to specify parameters for selected service and credentials
- After receiving file list, provide UI for selection
- ...then call WAILSAPI again to relay request

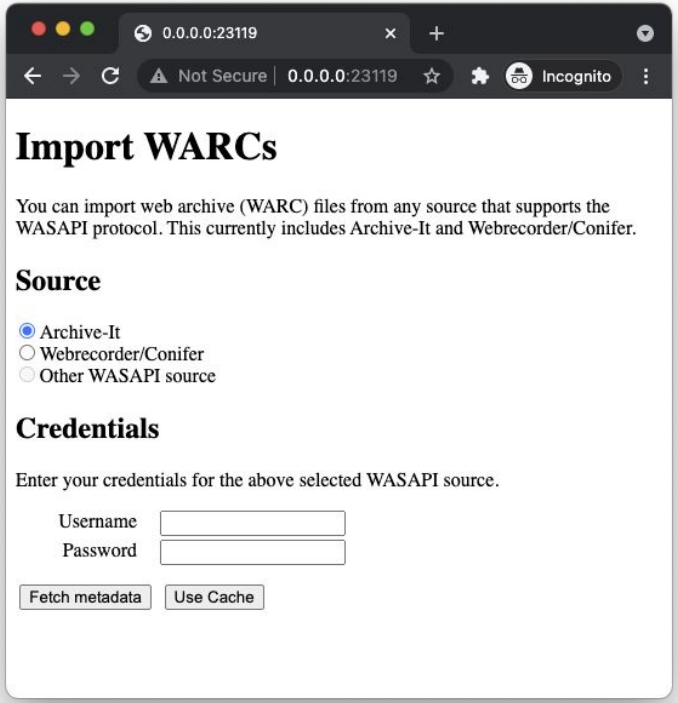


https://github.com/machawk1/wailsapi

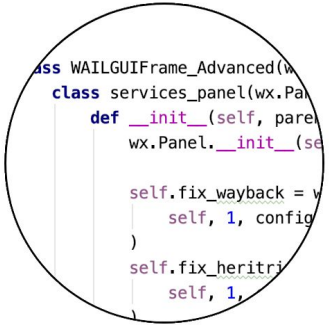
Bundled GUIs







Challenges (technical)



- Varying replay systems
  - e.g., OpenWayback ≠ Webrecorder's pywb ≠ ipwb replay
  - WARCs pulled from one source may not render correctly in different replay system
- Descriptive metadata for associated collection (e.g., collection name) requires secondary source beyond numerical identifier (e.g., 1234)
- Archive-It and Conifer's WASAPI server implementations are slightly different

Challenges (economical)

- Work described here is extending passion projects
  - Not driven by funding
- Questionably worthwhile endeavor as a faculty member
  - Both projects started while a grad student at ODU
- Funding might help justify progress, dev could be delegated to students
  - No longer passion projects
- Previously considered applying for IIPC discretionary funding
  - Ineligible, Drexel Univ. is not an IIPC member organization (yet!)
  - Progressed on project anyway

Progress

1. **Integrate WASAPI client into desktop tools**
2. Implement WASAPI server
3. Interface new clients to servers for additional data sources and WARC sharing (pull)
4. Extend to push (within WASAPI spec?)

True to the Title

WASAPIfying Private Web Archiving Tools for **Persistence** and **Collaboration**

- Interface as a client to Archive-It and Conifer/Webrecorder
- Extending Exist Desktop web archiving tools
- Facilitated by InterPlanetary Wayback (ipwb)
  - e.g., pull from Archive-It, push to IPFS
- Offload and interoperate with existing services
  - Allow others to also replay your captures using ipwb

Future Work

- P2P discoverability
- Potential further integration with Webrecorder stack
  - previously done with WAIL-Electron, unmaintained
  - cite our paper, show timeline relative to pywb development, provide github link
- Tighter coupling with py-wasapi-client
  - currently uses as means to identify WARC URIs
- Improve UI/UX to be more intuitive/easy-to-use

Try Them Out

https://github.com/machawk1/wail

- Native desktop application (primarily macOS)
- Also available on homebrew! `brew install wail`

https://github.com/oduwsdl/ipwb

- Link Instructions (separate repo)
- Link to Use Case

More info, these slides, etc: https://matkelly.com/iipcwac2021