Web-At-Risk
Report on Base Crawler Development

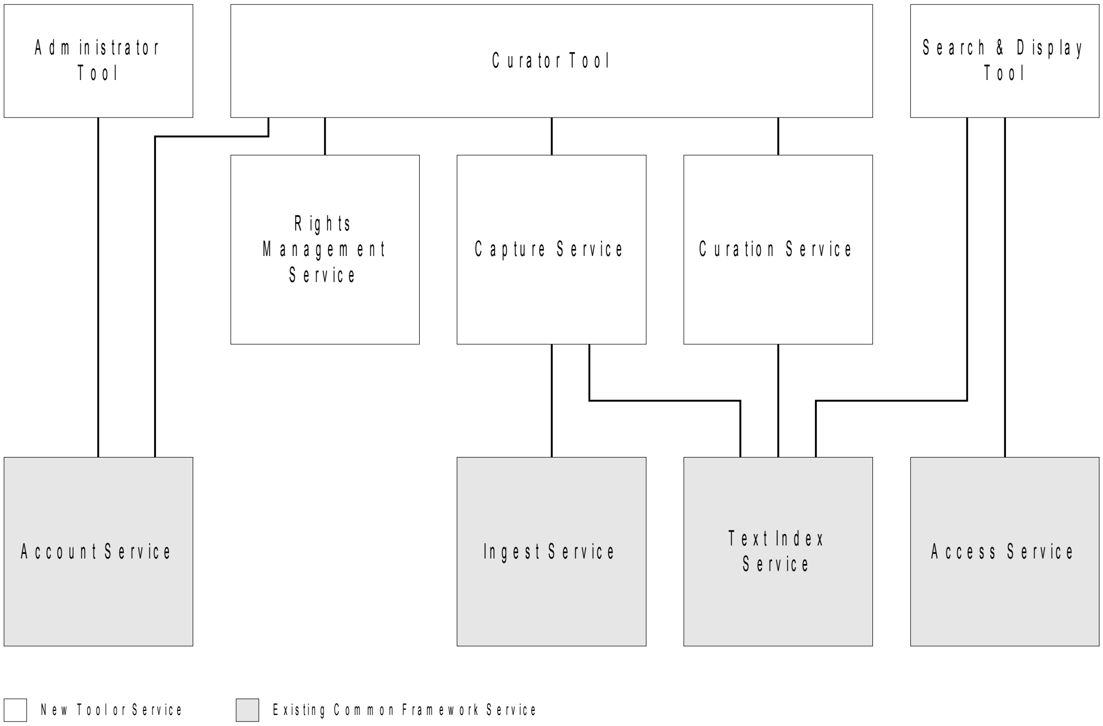
December 2005

This report summarizes the Web-At-Risk's work to date in and around the base crawler development at the heart of the CDL's Web Archiving Service. It covers the design work, our involvement in the forward-looking WARC file format evolution, and a prototype curator user interface that ties our envisioned services together to provide a reality-check on the systematic design work that is being done in parallel.

Design Status

The base crawler will reside at the core of our Web Archiving Service (WAS), the architecture of which is near completion as reflected in two major design documents[1][2] that represent an evolution of earlier modeling work[3]. The current high level architecture document serves as an introduction to a suite of related documents that describe individual parts of the proposed architecture in greater detail.

The following diagram illustrates the proposed WAS architecture in terms of new services and tools, and existing CDL Common Framework (CF) tools with which these new components would interact:



[1] WAS Architecture Overview, Draft 8, http://wiki.cdlib.org/WebAtRisk/tiki-download_file.php?fileId=96
[2] Capture Service Design, Draft 2, http://wiki.cdlib.org/WebAtRisk/tiki-download_file.php?fileId=58
[3] Web Archive Representation Model, http://wiki.cdlib.org/WebAtRisk/tiki-download_file.php?fileId=18

The proposed new services are:

Capture

A service whose core function is to pull content from remote locations on a one-time or repeated basis, and whose WAS-specific function is to pull web content on a one-time or scheduled basis via web crawling.

Curation

A service whose core function is to identify remote content to be captured, to annotate captured content and to organize captured content into collections. The specialized WAS version of this service will identify seeds for web crawls, annotate crawled content and organize crawled content into collections.

Rights Management

A service whose core function is to record rights information about possible targets for capture and for captured content.

Three tools will be developed that will deliver the functionality of these services to users by interacting with them through a user interface and communicating with multiple services through a set of API's. The tools are:

Curator Tool

A tool that will deliver to Curators the ability to capture, curate and preserve web content, as described in Section 3 of the WAS Requirements. Specifically, the Curator Tool will provide the following functions to the Curator:

- Login, via the Account Service
- Help
- Crawling, via the Capture Service
- Build Collections, via the Curation Service
- Manage Rights, via the Rights Management Service
- Generate Reports, via the Capture, Curation and Rights Management Services
- Preserve, via the Capture Service and (indirectly) the Ingest Service

Administrator Tool

A tool for Institution Administrators and Web Archiving Service Administrators (as defined in Section 1.3 of the WAS Requirements) to manage the WAS. Institutional Administrators will use the tool to:

- Login, via the Account Service
- Manage Accounts, via the Account Service
- Generate Reports, via the Account, Capture, Curation and Rights Management Services

Archiving Service Administrators will also have the ability to:

- Manage Institution Administrator accounts, via the Account Service
- Generate additional reports via the Account Service

- Monitor crawl schedules and crawl progress, via the Capture Service

Search and Display Tool

A tool that will deliver to all users the ability to search, browse and display preserved Web content. Preserved content will be retrieved from the CF repository via the Access Service.

Capture/Crawling Service

The Capture Service permits initiating and monitoring automated capture of content available on the network, whether the retrieved objects are OAI-PMH metadata or web content. This pulling of content from remote locations can either be scheduled for repetition, or can happen once only.

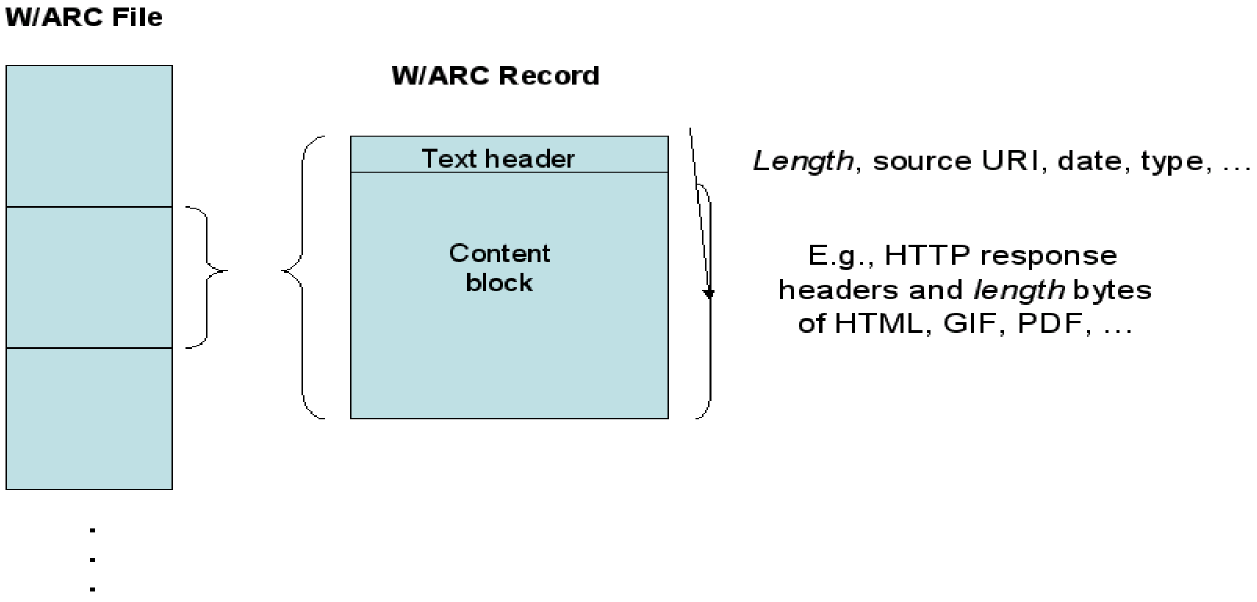
The Capture Service is deliberately generic so that metadata harvesting and other types of non-crawl capture may eventually be regulated by the same service. Although the Capture Service, and its subclass the Crawling Service, are designed for use with the CDL CF, this service should also be able to stand alone or interact with a different repository.

The Crawling Service permits initiating and monitoring web crawls and processing web crawl results. This service will interact with a new CDL CF resource manager, which in turn will interact with a capture agent which wraps third-party crawler instances (e.g., Heritrix).

WARC Status

The WARC (Web ARChive) is the next generation file format envisioned for use with the Heritrix crawler. This work was initiated by the IIPC (International Internet Preservation Consortium) to accommodate the new requirements that national libraries were bringing to in-depth crawling of their national webs. While this work is mature and has been slated for standardization within ISO, it is not yet complete so some of our modeling and primary crawler work is pending. Nonetheless, we can still anticipate the functionality that the WARC will bring compared to the ARC format upon which it is based. The CDL is actively participating in the ongoing WARC development.

The WARC format, like the ARC format, is very simple:


W/ARC File
W/ARC Record
Text header
Content block
Length, source URI, date, type, …
E.g., HTTP response headers and *length* bytes of HTML, GIF, PDF, …

WARC goals include:

- Ability to store arbitrary metadata linked to other stored data (e.g., subject classifier, discovered language, encoding)
- Support for data compression and maintenance of data record integrity
- Ability to store all control information from the harvesting protocol (e.g., request headers), not just response information.
- Ability to store the results of data migrations linked to other stored data
- Ability to store a duplicate detection event
- Sufficiently different from the legacy ARC
- Ability to store globally unique record identifiers
- Support for deterministic handling of long records (e.g., truncation, segmentation).

Base Crawler Infrastructure

Our base crawler infrastructure is designed to permit the fast, flexible, and fault-tolerant deployment of a dozen separate instances of Heritrix. These instances can be distributed, as needed, across several computers and storage networks in Oakland and Berkeley. We have kept up with current releases of Heritrix through three production releases.

The crawler itself will be integrated in our CDL CF technical infrastructure, dependencies for which are enumerated in[4]. This document was produced following our November 17-18 programmer retreat to identify for our partners all third-party open-source software required for installation of the CF and the Digital Preservation Repository (DPR) that uses it.

We are working with the San Diego Supercomputer Center to study crawler profile tradeoffs and to design a number of specific profiles designed for particular purposes. As an example, a specific profile might be used to focus on the in-depth capture of material from exactly one web site. Another might focus on retrieval of a given document with the strong assumption that all structural components of the document would share the path component with the entry point URL, but allowing an exception for any in-line images (images are commonly stored in a separate part of a website path hierarchy). Along these lines, we created our own “scope-plus-one” modifications to Heritrix because it wasn't possible to use profiles to express a capture of one entire site plus exactly one level beyond it (for all externally pointing links).

Prototype User Interface

The curatorial experience of the base crawler has been prototyped in a curator User Interface (UI) that consists of some HTML forms using PHP scripts and shell scripts to call Heritrix, then NutchWAX, and finally a modified version of WERA. We have used the UI to gain experiences with complete round-trip harvesting and indexing and to flush out bugs and hidden issues with the open source tools that we are still considering.

When a curator supplies an Entry Point URL, the UI asks Heritrix to begin crawling, and after a small delay NutchWAX is started to begin the simultaneous indexing of captured text-based content. When crawling is finished, and indexing finishes shortly thereafter, the curator is invited to search the crawled content. Capture and indexing is incremental in the sense that content accumulates over repeated crawls. Currently, the prototype supports only one collection and one user account.

[4] CDL Common Framework Prerequisites, http://wiki.cdlib.org/WebAtRisk/tiki-download_file.php?fileId=90

During the crawl, updated file, byte, and time counters are displayed via an HTML page that periodically refreshes itself. User inputs are validated to a limited extent; currently URLs are validated and an initial rights declaration is enforced. The interface also invites the entry of crawl-level metadata (but none of it is recorded). Various system conditions are checked and reported, such as when the crawler or indexer is unavailable. The user also has the ability to abort a crawl.

Source code and a demo of the prototype UI can be made available to the Library of Congress upon request.