Volume 40 | Number 4 Article 5

4-1-2013

myKive: Facilitating the Aggregation of Personal Archives

Chris Prom
University of Illinois, prom@illinois.edu

Electronic Currents—Joanne Kaczmarek, Assistant Editor, University of Illinois

MACers interested in writing a column for "Electronic Currents" are encouraged to contact Joanne Kaczmarek at jkaczmar@illinois.edu.

myKive: Facilitating the Aggregation of Personal Archives

By Chris Prom, University of Illinois

The myKive project proceeds from a simple working assumption: that one of the greatest challenges facing archivists in the digital era is the fact that the evidence about an individual's activities is scattered across many computers and servers. This problem is not wholly new. We all know that, when dealing with the physical papers of an individual, we often need to track down documentation from multiple rooms, buildings, cities, states, or even countries. The widespread use of networked computers (and in particular so-called cloud services) has exponentially increased what might be termed the documentation dispersion rate. The scattering of records may be virtual, but the effect is real.

To address this challenge, the myKive project is testing the feasibility of developing a free and open source software application that can be used to copy, preserve, and manage personal social media records, E-mail, and desktop files that are stored in disparate virtual locations. The project seeks to weave existing and/or newly developed open source tools into an integrated package of services that can be installed on a Web server and administered via a Web dashboard. The software would provide a concrete tool that archivists could use to engage with people who are creating records of potential archival value, but who are not yet ready to donate those records to an archival repository.

The proposed myKive service will allow people to push or pull records and information from E-mail accounts, blogs, Facebook pages, Twitter feeds, local computers, or other sources that are connected to the Internet or a local network. These records will be saved to a redundant, cloud-based server in a standardized, preservation-ready format. Regular integrity checks will be run against duplicate copies, and the checksums generated at time of packaging. Content will be stored with sufficient technical and structural metadata to permit its long-term preservation. Technically, the service will be based on very common Web technologies (Ruby on Rails, PHP, AJAX, REST, JSON) and preservation standards (OAIS, PreMIS). In providing these functions, the myKive project does not seek to replace a person's existing applications or to affect daily behavior, but rather aims to provide a method by which he or she can aggregate and make useful content that is currently scattered to the virtual winds.

As part of the myKive pilot project, the University of Illinois proposes to wrap four pieces of software into a Web-based dashboard application: (1) a social media archiving tool; (2) an E-mail archiving tool; (3) a desktop archiving tool; and (4) a visualization and mining tool. By focusing on three critical formats (social media, E-mail, and desktop files), the project targets format types that are most widely used by people in their personal and work lives. By providing visualization tools, the records will be immediately useful. We suggest that preservation be provided as a critical, but secondary, benefit to the end user. While users will certainly benefit from its preservation aspects, the project is based on the presupposition that the service will be much more likely to be used if it provides its users with an immediate benefit (the visualization tools) rather than a distant one (the preservation services).

As of January 2013, we have developed a provisional technical model and have tested application components. We anticipate that the application will use and extend existing open source software, which will be wrapped inside middleware to be developed by the University of Illinois Library. This middleware will link the individual system components into an overall "archiving" application, managed from a Web-based dashboard. The service core will consist of an application programming interface (API) that links the components, bundles them into a dashboard, and supplies back-end preservation services.

Within this overall framework, the proposed application will consist of the following elements, shown in schematic form (Figure 1):

- **myKive API**: This element will control user and account management, system security, and preservation packaging. The system will use a Model/View/Controller (MVC) framework to segregate application control and data views from the object model, which stores and manipulates information in user accounts, and the application logic, which determines how the data is updated, queried, stored, and viewed. As of January 2013, the basic modeling work for user registration and authentication has been completed.

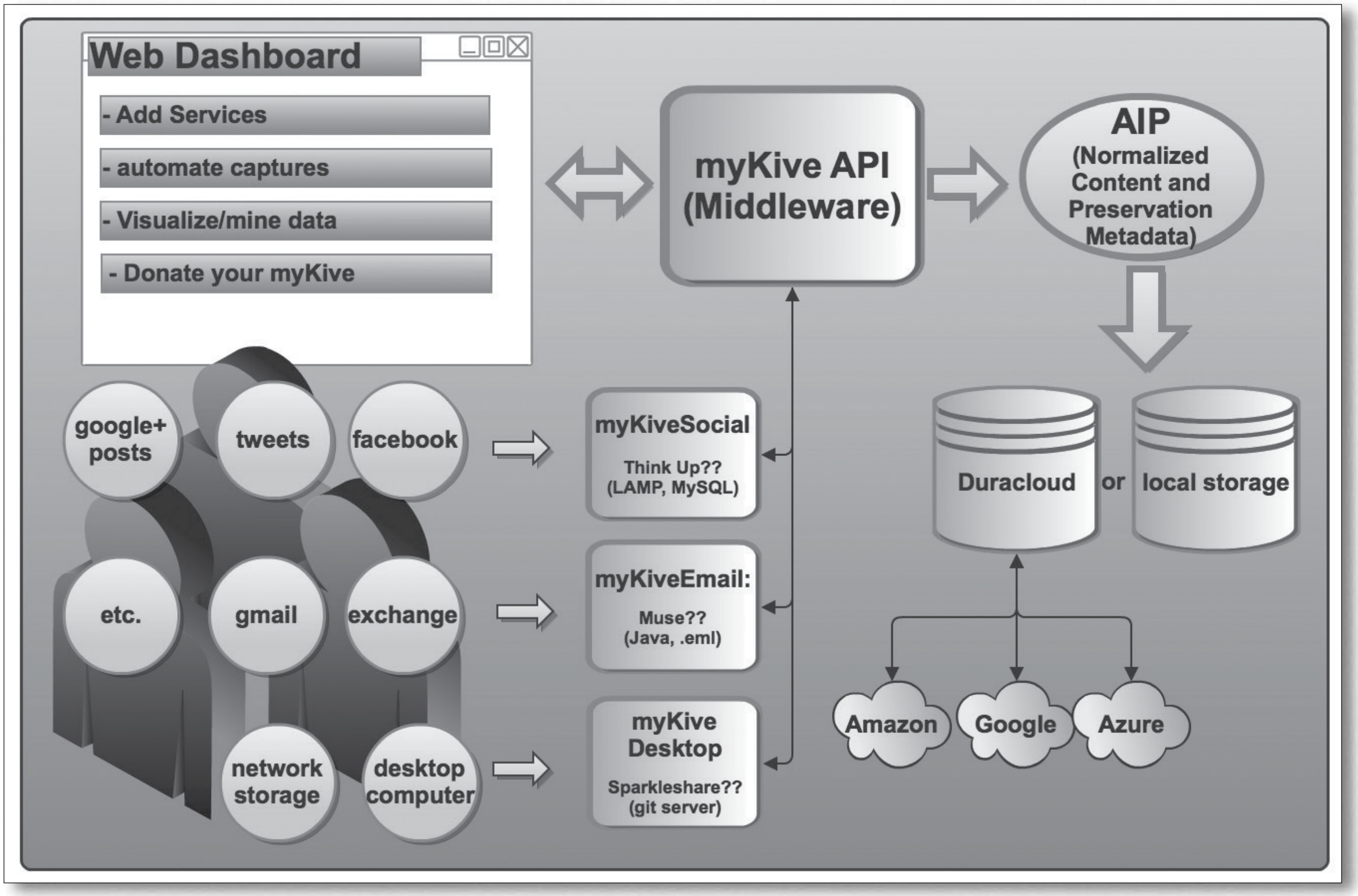

Figure 1: Proposed myKive Architecture (September 2012).

- **myKive Social**: When we developed the provisional project architecture, we suggested that the ThinkUp application be used as the basis for social media harvesting, with appropriate extensions and links to our application core.[1] ThinkUp, which is described as a "Social Media Insights Platform," provides people a way to harvest and aggregate tweets, Facebook content, and Google Plus postings, then to visualize and understand the ways in which social information is being produced, consumed, and repurposed. During the fall of 2012, we installed the tool and conducted preliminary testing. While it works well for its stated purpose, several technical factors (including the fact that the application lacks a theming system and an API) impede its effective integration into external services. In addition, it is not a preservation tool; the system does not capture some critical elements of social media posting (for example, uploaded photographs, audio, and video). As of January 2013, we are exploring whether changes in the beta version will remove some of these technical hurdles. Alternately, we will consider contributing to the overall project in a way that integrates preservation services into the core, without impeding the overall goals of the founding project developers. For example, we may develop a simple API that will allow external applications to register user accounts with ThinkUp.
- **myKive Email**: The open source MUSE software is suggested as a candidate technology for incorporation into myKive, since it uses an open storage format that facilitates data reuse and transformation, including visualizing, graphing, searching, and browsing tools.[2] Currently, MUSE is a JAVA extension to popular Web browsers, such as Chrome, Firefox, and Safari. It installs a JAVA applet on the user's computer. Once installed, it connects to a designated E-mail server, authenticates with user-supplied credentials, then downloads a copy of all sent and/or received messages to the desktop machine. MUSE provides visualization tools to make the E-mail searchable and more useful; for example, users can view E-mail by sentiment, topic, or user-defined criteria. We propose utilizing the MUSE core

(Continued on page 16)

(Continued from page 15)

and building a server-based version of the software. We will complete initial testing of the application during the summer of 2013 to determine if it is suitable for this purpose.

- **myKive Desktop**: We propose using secure socket layer technologies and standard encryption techniques to capture and store desktop files from a local computer on a synchronized and replicated server. The SparkleShare application is suggested as a candidate technology for testing.[3] Sparkleshare is an open source alternative to file sharing and backup applications like Dropbox and Carbonite; it uses Git as its back-end architecture.[4] Our hypothesis is that we can establish a Sparkleshare server and then integrate the Sparkleshare synchronization tools into our myKive application core/dashboard in a way that allows a user to sync a designated computer to the back-end Git repository. We hope to add automated desktop file backup, synchronization, and even version control into the system. In theory, users will be able to choose whether to keep a complete record of the desktop over time, or only the latest version of the files.
- **Dashboard with Visualization Tools**: Users will be provided a dashboard application to manage the system components described above. Once they complete initial setup, users will need to perform little or no maintenance, other than keeping passwords current and authenticating from time to time, as required by the terms of the social media APIs on which the service depends. By providing visualization and data-mining tools, such as those integrated into MUSE and ThinkUp, we believe users will have a reason to remain interested in and use the application.

One element of the service is not shown in the schematic diagram: the core API will include a plug-in architecture so that the service can preserve other social media objects. This will allow people from the computer science, archives, library, and digital curation communities to extend the application. Using those extensions and plug-ins, people will be able to harvest records from other services into their myKive. For example, the following record types might be included in an individual's myKive, once appropriate extensions are developed:

- Blogs, using backup tools such as ArchivePress and WordPressDatabase Backup;
- Photographs, using capture tools such as parallel-flickr;
- Web pages, using harvesting tools such as wget, warc, and NutchWax; and
- Personal reference/citation libraries, using tools such as Zotero's application programming interface.

At this time, myKive is a pilot service in the very early stages of feasibility testing, using funding from the University of Illinois Library Innovation Fund.[5] The staff includes Project Director Chris Prom and System Engineer Vishnu Muthukrishnan, a computer science graduate student at the University of Illinois at Urbana-Champaign. As the technical model is developed, external partners will be sought to engage in pilot testing. If successful, we hope the project will garner widespread interest and that it may serve as the basis for a collaborative international project, with significant external funding and a long-term sustainability/business model.

Notes

1. *ThinkUp*, https://thinkup.com/ (5 February 2013).
2. Stanford University, Mobisocial Laboratory, *Muse,* http://mobisocial.stanford.edu/muse (5 February 2013).
3. *Sparkleshare*, http://sparkleshare.org (5 February 2013).
4. Git is an open source revision control system, typically used for source code management in open source projects. However, it can be used as part of any distributed file-sharing project. See *Wikipedia*, s.v. "Git (software)," http://en.wikipedia.org/wiki/Git_%28software%29 (5 February 2013).
5. Current information is available at the myKive Web site. See *myKive*, http://www.mykive.org (5 February 2013).