

Zotero Everywhere: Browser- and Machine-Independent Support for the Research Community

A Proposal to the Andrew W. Mellon Foundation

Center for History and New Media
George Mason University
Sean Takats & Dan Cohen

8 June 2010

The Past, Present, and Future of Zotero

Since the Andrew W. Mellon Foundation generously funded a major expansion of Zotero from its prototype and 1.0 phase to its current major 2.0 release, the project has established itself as the world's favorite open-source reference manager and digital research platform. Since its inception, Zotero has swept the globe with over four million downloads, with translations of its interface into over 40 languages by its dedicated user community. Zotero regularly receives accolades from scholarly societies (e.g., the American Association of Political Scientists' "Best Software" award) and the technology world (e.g., *PC Magazine's* "Best Free Software" recognition).

Through initiatives such as optional fee-based cloud storage through the recently formed Corporation for Digital Scholarship, a non-profit established to underwrite Zotero's continuing costs, the project has made major progress with respect to sustainability within the bounds of existing grant funding. Nonetheless, a critical challenge to Zotero's long-term sustainability remains its exclusive tethering to the Firefox browser. We therefore propose a staged project to decouple Zotero from Firefox by developing a standalone Zotero client, by creating new Zotero toolbars for other popular browsers, and by significantly expanding the Zotero server's functionality to support that client and other tools that interact with the platform. In short, to create an ecology that supports research wherever and however that research is being done.

An independent version of Zotero will have two chief benefits encouraging sustainability: user choice and a diverse—and thus sustainable—ecosystem of Zotero tools. A standalone Zotero client will give users the freedom and flexibility to use browsers other than Firefox—including browsers and applications on mobile platforms. Since the 2006 launch of Zotero, Firefox has grown fivefold in popularity, from under 5% to 25% of the total browser market. Although we continue to believe that our initial choice of platforms was a good one and that Firefox is (and will remain) an excellent web browser, the other major open-source browser, Google Chrome,

has been rapidly rising in popularity, quickly achieving a 7% market share. Apple's Safari has slowly but steadily cultivated 5% of the market. And of course Internet Explorer remains popular even after a decline to just under 60% of the usage share. In the face of this diversity, the major user complaint about adopting Zotero is the perception that it requires the use of a specific browser, Firefox. In some cases the user cannot even adopt Firefox as a secondary browser, since many institutional environments prohibit the use of alternative web browsers. Moreover, if Zotero is perceived as impeding user choice, even existing users may also be more inclined to abandon it in favor of standalone desktop software which is browser-agnostic.

Beyond the issue of user perception related to adoption and retention, a standalone version of Zotero also protects the project against any risks associated with Firefox's future. So long as Firefox has continued to gain market share at the expense of Internet Explorer, the rising tide helped to lift Zotero. But as Zotero has matured into software relied on by millions of users, we believe it is critical that Zotero not wholly depend on the future of Firefox. There is no good way to predict Firefox's medium- and long-term popularity, but recent data suggests that Google Chrome is slowing Firefox's growth in market share.¹

Fortunately, whatever the future brings for Firefox, the bulk of Zotero's codebase is completely secure, since we now believe it can be compiled as a standalone application. We have already begun work on a proof-of-concept that demonstrates just that. Moreover, Zotero's server-based code already provides limited support for other browsers and mobile applications, and we propose to expand that functionality dramatically. Through its publicly available application programming (API) interface, the Zotero server will thus also mitigate against the risk of shifting browser market shares and the accelerating move toward mobile interfaces.

Because of its similar extension architecture, we were already confident in our ability to extend Chrome to support Zotero. Thanks to recent developments we are well-positioned to offer Zotero to other browsers as well. To ensure the feasibility and progress of the browser support, we have been in talks with Donald Brinkman and Alex Wade, external research project managers for Microsoft Research who have been following the progress of Zotero and are in charge of interoperability of open source software with Microsoft's products. Donald and Alex have offered the Zotero Project access to engineers in the Internet Explorer division of Microsoft (as well as to the Office division for future updates to our Word plugin), and they are also exploring the possibility of some in-kind labor to assist with Internet Explorer integration. We have also now examined the new extension documentation for Safari 5, which was released on June 7, 2010, and believe that it is indeed possible to add a Zotero toolbar to Apple's browser.

To bring this independent version of Zotero to market would require project funding that is well beyond what we are currently taking in through our sustainability plan, which for the foreseeable future is focused on continuing costs rather than major new developments such as this project. We are therefore proposing to the Mellon Foundation a targeted grant to produce this important new code.

http://www.readwriteweb.com/archives/firefox_losing_early_adopters_to_chrome.php

Changes to the Zotero Server and Client

Expanding the Zotero server's functionality will not only liberate Zotero from its dependence on Firefox, it will free Zotero users from relying on any specific machine to access the full functionality of Zotero. At present, users can access their Zotero library items and attached files via Zotero's web interface by visiting <http://zotero.org>. This access is currently read-only, but we propose to extend the server's functionality to include write access. As a result, Zotero users will be able to add new Zotero items from any web browser, without needing to install Zotero. They will moreover be able to edit or delete items from their Zotero libraries. Because the zotero.org website's functionality is all based on a publicly available application programming interface (API), third parties will also be able to develop applications that interact with Zotero data.

To facilitate the addition of Zotero items without needing to install the Zotero client, we propose to offer any Javascript-enabled web browser much of Zotero's groundbreaking "translation" functionality, which seamlessly extracts the structured metadata about scholarly objects from heterogeneous web pages. Via a lightweight bookmarklet, any browser user will be able to add items to their Zotero library without needing to install the Zotero software. This feature will also open the door to third-party applications, which will benefit from the significant investment of resources represented by Zotero's translator code, presently able to process tens of thousands of scholarly web resources and to archive snapshots of any web page. We expect that this service will further increase community interest in maintaining this vast and important codebase.

Beyond the ability to add, edit, and delete Zotero items via the web, we also will port Zotero's powerful search and citation rendering functionality to the server. Currently, PDFs, notes, and other Zotero text is only searchable from within the Zotero client, not on the Zotero server. By adding server-side search, we will dramatically improve access to Zotero data, both through the zotero.org website and programmatically via the API. Furthermore by adding a server-side version of Zotero's powerful citation rendering software, we will be able to provide a universally accessible interface for formatting references and bibliographies. Without any need to have Zotero installed—and indeed without even needing a Zotero library—users will be able to pass bibliographic metadata to the Zotero server and receive properly formatted citations.

The project is not a small one and poses some challenges. It will require major rewriting of some core components and parallel maintenance of code for many different iterations of browsers and platforms. Standalone Zotero and the browser plugins will require cross-platform development and testing. Porting of components to the server may require significant rewriting and optimization of the code. We expect some wrinkles along the way and have built into the timeline additional time to deal with unexpected browser updates or changes, bugs and unforeseen compatibility issues, and the additional support burdens for the community during the phased launch.

However, we are confident that there is minimal theoretical work required for this project and that it is mostly a (perhaps at times complex) implementation project. We are also aware that we

are moving from a fairly clear descriptive environment—"Zotero is a Firefox plug-in for research"—to a broader project. Differentiating various types of Zotero will pose a marketing challenge, even though the goal is to provide users with the maximum amount of choice and flexibility. For this reason we include in this proposal substantial work by our community lead, who will do outreach and support for the now varied set of users and their platforms, which will be especially necessary in this first wave of releases. Ultimately, however, Zotero users will greatly benefit from the ability to interact with their research data with the same degree of flexibility that users now enjoy when accessing email: from a personal computer, a lightweight web interface, and from mobile devices.

Project Deliverables

1. *Independence from Firefox*

- a. Standalone Zotero compiled and running on Windows/Mac/Linux. Zotero will be compiled into an application that can run on any of operating systems supported by Firefox now or in the future. Thanks to Mozilla's XULRunner project, Firefox add-ons can be compiled into software that does not require the Firefox browser. Basic compiling of a functional standalone version of Zotero thus requires relatively little work, but some aspects of Zotero will need to be adjusted in order to accommodate a standalone environment. For example, some Zotero functionality, like creating a web snapshot with the click of a button, will no longer make sense in the context of a standalone application. Therefore we will need to tweak the Zotero user experience to reflect the different capabilities of standalone Zotero versus the traditional Zotero Firefox add-on.
- b. New browser plugins. To interact with the new translator architecture described above, we will prepare browser plugins for several other popular web browsers. These plugins will provide the standalone Zotero application with all the information it needs to sense whether there is meaningful data that can be extracted from the page by passing a URL, the page source, and any relevant other data. If Zotero detects a page that can be translated, it will then notify the browser plugin, which will display an appropriate icon for the user to click, should the user wish to add this resource to her Zotero library. We will provide plugins for Google Chrome, Microsoft Internet Explorer, Apple Safari, and Mozilla Firefox, which combined account for 98% of the usage share of web browsers.
- c. A new site translator architecture that is decoupled from the Firefox browser. The key to Zotero's interoperability with a wide range of digital resources is its innovative "translator" framework that allows Zotero to extract meaningful data from heterogeneous web pages by identifying embedded metadata standards or through code specifically written for individual websites and services, both open and gated. These code snippets, created using Javascript and XPath, effectively render the web semantic without additional work by the holders of

scholarly collections. Zotero's translators currently only run locally, within the Zotero Firefox add-on, where Zotero has full access to the structure and data of web pages being viewed by Zotero users. In order to bring Zotero to other browsers, Zotero will have to provide a browser-independent translator architecture that allows an arbitrary web browser to supply information about a web page to Zotero, and for Zotero to process that information.

2. *Machine Independence*

- a. Syncing of full-text content. We will add syncing of full-text content to allow users to access their attachments (e.g. PDFs) from any installation of Zotero or via the web at zotero.org. This full-text content will be indexed and searchable from any of these access points, allowing users to find items in their library via keyword search. With full-text content available, we will also be able to extend Zotero's powerful saved search functionality to zotero.org. Just as users can currently create a virtual Zotero collection on their personal computers through the assignment of various search criteria, they will now also be able to access those saved searches at zotero.org.
- b. Expansion of the Zotero Server API. At present the Zotero Server API provides read-only access to personal and group libraries. As a result, Zotero users and third parties can programmatically access Zotero data to build a variety of applications or to perform computational analysis. We will add the necessary functionality for users and third parties to make changes to Zotero data via the API, including the ability to add new items, edit existing items, and delete items. This new API functionality will provide the essential foundation for third parties to develop full-featured applications interacting with Zotero data, including mobile applications.
- c. Library editing via the zotero.org website. Using the expanded API described above in 2b, we will produce a vastly improved web interface that will allow users to add, edit, and delete items. We will also for the first time permit server-side tagging of single and multiple items and the organization of Zotero items into collections. All of these changes will propagate to a user's client(s) through the sync process. Zotero's current sync functionality allows users to edit their Zotero libraries on any machine with Zotero installed. With the new web interface, they will be able to edit their libraries via any web browser.
- d. Adding Zotero items via a bookmarklet. Zotero's translator architecture is currently tethered exclusively to the Zotero Firefox add-on. Only the Zotero client software can sense research content and extract metadata. We propose to offer this functionality on the Zotero server as well. This simple way—simpler even than a toolbar—to add items to a user's Zotero library will require us to create bookmarklets (bookmarks with Javascript in them to pass information to a web service). As a result, users will have the ability to add new items to their Zotero

libraries without manually entering bibliographic data, just as they already can with Zotero installed. For instance, this could be used by a developer creating an iPhone, iPad, or Android app to pass new additions to the Zotero server.

- e. A generic citation API. Zotero 2.1 currently integrates an entirely rewritten citation formatting engine which can also run as a standalone server module. We propose to install this module on the Zotero servers and to make its functionality accessible via the API. Developers will need only to append the unique identifier of one of the thousands of publication styles supported by Zotero to their API requests, and the Zotero servers will return a properly formatted citation or bibliography reference.
- f. On-demand file download. Even users with very large libraries will be able to access their data from locations with limited bandwidth, since there will be no need to synchronize all file attachments. This functionality will also be especially interesting to users on mobile devices, facing bandwidth restrictions, or who otherwise lack high-speed access to the internet.

Sustainability

Through the non-profit Corporation for Digital Scholarship, the Zotero Project is able to offer optional additional storage for users of the Zotero server who wish to back up their files online and share them in groups. Over a thousand people have signed up for additional storage through CDS in the first few months, and these paid-yearly funds will be used to support personnel and technical costs for the long-range sustainability of the program. We expect that with ongoing renewals and new purchases of storage by individuals and institutions that at the end of this grant Zotero will be self-sustaining. Currently we are running at about a quarter of the necessary yearly funding needed through CDS, so additional users garnered through this grant project will be an enormous help, as will institutional sales enabled by non-Firefox browser support. The project believes it can run on as little as 1% paid users of the larger user base we expect at the completion of the Zotero Everywhere project.

Milestones (assuming start date of October 1, 2010)

Desktop Milestones

Alpha 1 release (January 2011): Standalone Zotero (Kornblith, Stillman, and Owens) compiled and tested on Mac OS X, Windows, and Linux

- All browser-specific functionality removed
- Unit-testing to guarantee standalone functionality
- Translator architecture isolated from Firefox
- Browser-independent transmission of page data to Zotero
- Release of the first browser plugin, for Google Chrome

- Initial drafts of new documentation

Alpha 2 (February 2011): Firefox Support (Kornblith)

- Release of the second and third browser plugins, for Apple Safari and Mozilla Firefox

Alpha 3 (April 2011): Internet Explorer Support (Kornblith)

- Release of the fourth browser plugin, for Internet Explorer 8.

Public Beta (June 2011) (Owens, Kornblith, and Stillman)

- This first official public launch of the standalone client and new toolbars will help us to isolate issues with particular platforms and browser versions.
- New support sections of the website will encourage bug reporting and assist in troubleshooting.
- At this stage we also expect to be actively working with our community of external developers to help with sustainability of the toolbars for the long run (e.g., Internet Explorer 9, 10).

Production stage (September 2011) (Owens, Kornblith, and Stillman)

- Production version of Standalone Zotero publicly available
- Final 1.0 versions of toolbars

Iterations and transition (January - September 2012) (Owens, Kornblith, and Stillman)

- Continuing bug fixes and support for expected updated and new versions of supported browsers
- Solidify developer and user communities around each toolbar and Standalone Zotero

Server Milestones

Zotero API v. 3.0 (December 2010) (Stillman)

- Zotero public API support for add, edit, and delete

Documentation for Expanded API (March 2011) (Owens)

- Online documentation for expanded API functions (add/edit/delete)

Zotero Web Interface v. 3.0 (March 2011) (Cheslack-Postava, Raymond)

- Web interface supporting the new API functions

Documentation for Web-Based Library Use (March 2011) (Owens)

- Video tutorials and written documentation expanded to reflect web-based interaction with library data, not just Zotero client interaction

On-Demand File Download (April 2011) (Stillman)

- Users download individual files into Zotero without needing to synchronize the entire library
- API functionality exposed for mobile applications, as well

Zotero Web Interface v. 3.5 (August 2011) (Cheslack-Postava, Raymond)

- Optimized for mobile browsers (Android, iPhone, iPad)

Documentation for Mobile Library Use (August 2011) (Owens)

- Video tutorials and written documentation expanded to reflect mobile device interaction with library data

Citation API (November 2011) (Stillman, Cheslack-Postava)

- Formatting service and public API will allow any application to format Zotero items

Bookmarklet (February 2012) (Kornblith)

- Any browser can add Zotero items automatically.

Full-Text Everywhere (March 2012) (Stillman, Cheslack-Postava, Raymond)

- New client-side and server-side indexing and search code
- Web site search interface

Iteration support and upgrades (March - September 2012) (Stillman, Cheslack-Postava)

- Ensure continued effective interfaces with desktop components, especially for new desktop and mobile browser versions
- Ensure interoperability with independently-developed citation processing engine

Ongoing tasks

- Technical support for new API (Stillman)
- Web application maintenance and support (Shepherd, Cheslack-Postava)
- Community outreach and management of alpha and beta feedback (Owens)

Personnel

Sean Takats and Dan Cohen, Co-Directors

Dr. Takats and Dr. Cohen will be responsible for the day-to-day oversight of the project as well as the strategic planning of the launch and integration of new code. They will also maintain relationships with partners and external developers critical to the project.

Dan Stillman, Lead Developer

Overall responsibility for the technical side of the project, including both client and server.

Faolan Cheslack-Postava, Web Developer

Responsibility for changes to the Zotero.org website, integration with other browsers.

Simon Kornblith, Integration Developer

Responsibility for independent translator architecture and browser plugin development.

Trevor Owens, Community Lead and Junior Web Developer

Will act as a go-between with the developers and the new browser communities. Also will set up new forum sections for feedback and troubleshooting of the inevitable problems for the non-Firefox browsers. New workshops for institutions implement enterprise versions, who are likely to be standardized on IE (yr. 2). New documentation of standalone client and installation and use of browser toolbars. Redo existing intro videos to be browser-agnostic.

Ammon Shepherd, System Administration

Responsibility for the technical infrastructure of the Zotero server cluster, including blades and switches. Also responsible for database administration and the maintenance of software libraries and programming languages for the website and web services for sync and the Zotero API.

Chris Raymond, Web Designer

User interface and user experience design of the Zotero website, Zotero clients, and browser-specific toolbars.

Andy Privee, Grant Administration

Runs the financial and office management tasks for Center for History and New Media projects, which is especially important for distributed projects such as this one.

Budget

Please see the budget spreadsheet, sent separately.