



UNIVERSITY OF CALIFORNIA, BERKELEY  
SCHOOL OF INFORMATION

# Bridging the Gap between eBook Readers and Browsers

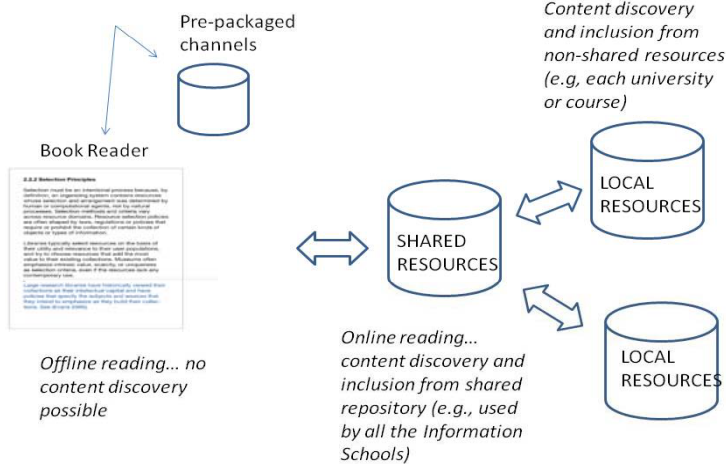
[eBooks: Great Expectations for Web Standards](#)  
[W3C Workshop on Electronic Books and the Open Web Platform](#)

Robert J. Glushko ([glushko@berkeley.edu](mailto:glushko@berkeley.edu))  
Erik Wilde ([dret@berkeley.edu](mailto:dret@berkeley.edu))  
Luis Aguilar ([luis@ischool.berkeley.edu](mailto:luis@ischool.berkeley.edu))  
Fred Chasen ([fchasen@ischool.berkeley.edu](mailto:fchasen@ischool.berkeley.edu))

February 11, 2013

## One (of many) Motivation(s): Collaborative Teaching with a [Multidisciplinary Textbook](#)

### A DEPLOYED EBOOK



## Books and the Web

- “On” the Web vs. “of” the Web
  - “on the Web” means Web as a delivery channel
  - “of the Web” means Web as the service fabric
- Think “Flash sites” or “Google docs” vs. “Web sites”
  - The first two require proprietary client-side support
  - Web sites require only standards client-side
- Current ebook readers are closed devices
  - Consuming machines within a walled content garden
  - Maybe some reuse of web technology, but primarily for engineering reasons (e.g. webkit for rendering)

3

## A Browser is a Reader A Reader is a Browser

- Making eBooks *Web content* is essential
  - Prevents lock-in, promotes client diversity, open access
  - Content standards (HTML/CSS) are more than good enough
- Making eBooks *Web resources* is essential as well
  - eBooks should be first-class (linkable) Web citizens
  - Resource standards (URI/HTTP) should be followed
- Open access and service innovation are critical
  - Current packaging and distribution makes user-defined book libraries impossible, allowing only publisher and platform-based ones
  - Closed architectures should be left to industry for market segments that need and demand them

4

## Example: Storing Content

- Our scenario requires a uniform browser-agnostic solution for offline local access to resources
  - Resources on the Web are identified by URIs
  - Offline mode should provide offline access to URIs
- Many “Web technologies” are not very webby
  - Resource access requires much more than a URI
  - localStorage is a primitive client-side key/value store
  - File API provides access to the client’s file system
- If it doesn’t have a HTTP URI, it’s not on the Web
  - Must respond to the Web’s uniform interface
  - Interactions can be negotiated by the Web’s fabric, not by APIs that create a parallel universe

## AppCache Problems

- Designed for Web apps and not Web content
  - Packaging of HTML/JS/CSS/img for delivery
  - One “app” from one “store” to “run” in the browser
- What’s wrong with this picture?
  - Browser UIs for managing appcache are bad
  - Browser APIs for managing appcache do not exist
- Apps have fixed boundaries, ebook content is fuzzy
  - Apps often have a well-defined set of resources
  - ebooks are hypertext and can transclude the Web, so we need to be able to dynamically add content to appcache

6

## ConCache (or, “Intelligent Persistence”)

- Fix appcache to make sense for Web content
  - Add API for exploring and changing the client cache with URIs, not by filesystem or db methods
  - Defines a clear book “boundary” so you don’t cache the entire web
  - Both server and client changes should be possible
- Browsers should have decent UIs for handling content saved locally
  - Allow users to move content between browsers/computers
  - Allow users control over when it’s refreshed
  - Export/import could be something like ePub

7

## OS-ifying the Browser

### HTML5 Landscape Overview

HTML5 is more a movement (or maybe it's more appropriate to call it a "brand") than it is a technology. It says "more power to the browser" and mostly means "more power to the browser as a programming platform". Given this focus of HTML5, it is surprisingly hard to find a good place where all the APIs under development are listed. This page is an attempt to collect that information in one place. The current status captured on this page lists 61 current specifications and 4 expired specifications (65 total). Since the HTML5 landscape is changing fairly quickly, it is likely that some information on this page is outdated. If that is the case, please [get in touch](#) so that I can update the page. Thanks!

Spec Type	Specs
(none)	Clipboard API and Events <sup>WD</sup> Fullscreen <sup>WD</sup> HTML Microdata <sup>WD</sup> HTML5 Web Messaging <sup>CR</sup> High Resolution Time <sup>REC</sup> IndieUI: Events 1.0 <sup>WD</sup> Media Capture API <sup>WD</sup> MediaStream Processing API <sup>WD</sup> MediaStream Recording <sup>WD</sup> Messaging API <sup>WD</sup> Navigation Timing <sup>REC</sup> Navigation Timing 2 <sup>WD</sup> Page Visibility <sup>CR</sup> Performance Timeline <sup>CR</sup> pick Media Intent <sup>WD</sup> pointer Lock <sup>WD</sup> Programmable HTTP Caching and Serving <sup>WD</sup> Proximity Events <sup>WD</sup> Resource Timing <sup>CR</sup> Timing Control for Script-Based Animations <sup>WD</sup> URL <sup>WD</sup> User Timing <sup>CR</sup> Web Alarms API <sup>WD</sup> Specification <sup>WD</sup> Web Intents <sup>WD</sup> Web Intents Addendum — Local Services <sup>WD</sup> Web Notifications <sup>WD</sup> Web Workers <sup>CR</sup> WebDriver <sup>WD</sup>
Communications/Networking	Network Information API <sup>WD</sup> Push API <sup>WD</sup> Server-Sent Events <sup>CR</sup> WebRTC 1.0: Real-time Communication Between Browsers <sup>WD</sup> WebSocket API <sup>CR</sup> XMLHttpRequest <sup>WD</sup> XMLHttpRequest Level 2 <sup>WD</sup>
DOM	DOM Parsing and Serialization <sup>WD</sup> Selectors API Level 1 <sup>WD</sup> Selectors API Level 2 <sup>WD</sup>
File System	File API <sup>WD</sup> File API: Directories and System <sup>WD</sup> File API: Writer <sup>WD</sup>
Graphics	HTML Canvas 2D Context <sup>WD</sup>
Hardware	Ambient Light Events <sup>WD</sup> Battery Status Event Specification <sup>CR</sup> DeviceOrientation Event Specification <sup>WD</sup> Gamepad <sup>WD</sup> Geolocation API <sup>WD</sup> Specification <sup>CR</sup> Network Service Discovery <sup>WD</sup> Pointer Events <sup>WD</sup> Screen Orientation API <sup>WD</sup> Touch Events Version 1 <sup>WD</sup> Vibration API <sup>CR</sup>
Media	Audio Processing API <sup>WD</sup> HTML Media Capture <sup>WD</sup> Media Source Extensions <sup>WD</sup> Web Audio API <sup>WD</sup> Web MIDI API <sup>WD</sup>
OS-Environment	Calendar API <sup>WD</sup> Contacts API <sup>WD</sup>
Security	Web Cryptography API <sup>WD</sup> WebCrypto Key Discovery <sup>WD</sup>
Storage	Indexed Database API <sup>WD</sup> Quota Management API <sup>WD</sup> Web SQL Database <sup>WD</sup> Web Storage <sup>CR</sup>

<http://dret.typepad.com/dretblog/html5-api-overview.html>

8

# Thanks!

Robert J. Glushko ([glushko@berkeley.edu](mailto:glushko@berkeley.edu))

Erik Wilde ([dret@berkeley.edu](mailto:dret@berkeley.edu))

Luis Aguilar ([luis@ischool.berkeley.edu](mailto:luis@ischool.berkeley.edu))

Fred Chasen ([fchasen@ischool.berkeley.edu](mailto:fchasen@ischool.berkeley.edu))

9