

Mozilla Prism Anant Narayanan Gnunify 2008

In the old days...

- * You open a web browser, check your email, log off and close it
- * Few people had 24x7 internet connectivity
- * Web applications were less "persistent"

Today

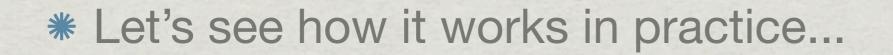
- * Applications like GMail and Google Docs are serious alternatives to their desktop counterparts
- * The number of people who keep their browser windows open all day are increasing
- * Web applications are becoming more persistent, and are offering users a more responsive interface

What is Prism

* Suitable for running persistent web applications that you expect to run a long time

- Instead of keeping your browser open, run the web application just as if it were a desktop application
- * Basically a browser, but without all the unnecessary chrome since you're only going to be using a particular application

Demo



Why prism can be useful

* In it's current state:

- * Quick access to your most frequently used web applications
- * Close integration with your desktop

Comparisons

* Prism is frequently compared with AIR, Silverlight, JavaFX and Fluid

- * Prism leverages existing web technologies to provide the user with a traditional desktop application experience
- * Let's take a look at the "competition"

AIR

* Adobe Integrated Runtime

* Leverages the Flash/Flex platform to deliver desktop applications that can be built by web developers

* Requires a client-side application to execute the application (like XULRunner!)

Silverlight / Moonlight

- Silverlight: Microsoft's browser plugin Moonlight: Mono's open source implementation
- * All about providing rich user interfaces, much like flash does
- * Targeted primarily at the web, no current plans to allow desktop-only execution



Java pioneered the concept of "write-once, execute everywhere"

* Applets have always been able to run as standalone desktop applications

JavaFX is about "rich internet applications" - it targets only the web as of today

Fluid

* A more direct competitor to Prism

* OS X (Leopard) only application that is pushing for even more desktop integration

* Based on Webkit

* Prism is trying to add features that Fluid already provides, but do so in a web-safe way

Where does Prism stand

- * Existing web applications work with Prism out of the box
- * Further integration to the desktop may be done without needing the web application developer to do anything
 - * (May not be possible for features like offline storage support)

The future

* Provide offline event and storage support

- * Provide more power to the web application developer by providing access to hardware capabilities of the client
- * Allow web applications to register handlers for certain content types
- * These features are being standardized in HTML 5!

More Ideas

* Provide UI capabilities that match desktop applications

* Drag and drop directly onto web applications

* Your idea - Make sure it is heard!

Get Involved!

* Report user experiences, which web applications work well with Prism and which do not

* Report problems, Discuss Features

* Forum: <u>https://labs.mozilla.com/forum/index.php/board,</u> <u>16.0.html</u>

* Check out the code (Mozilla SVN - Webrunner)

* Fix bugs! (Prism component in bugzilla)

Questions?

Thank You!