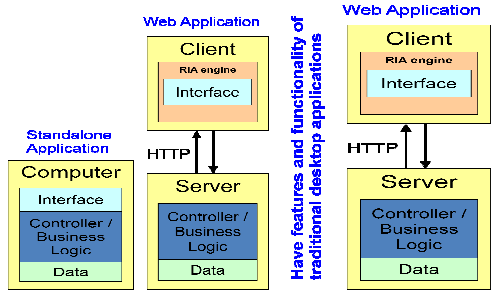
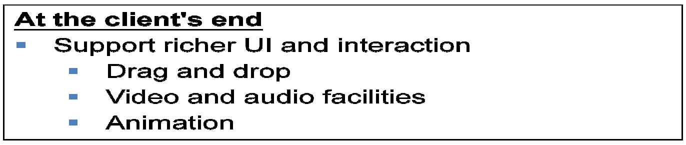
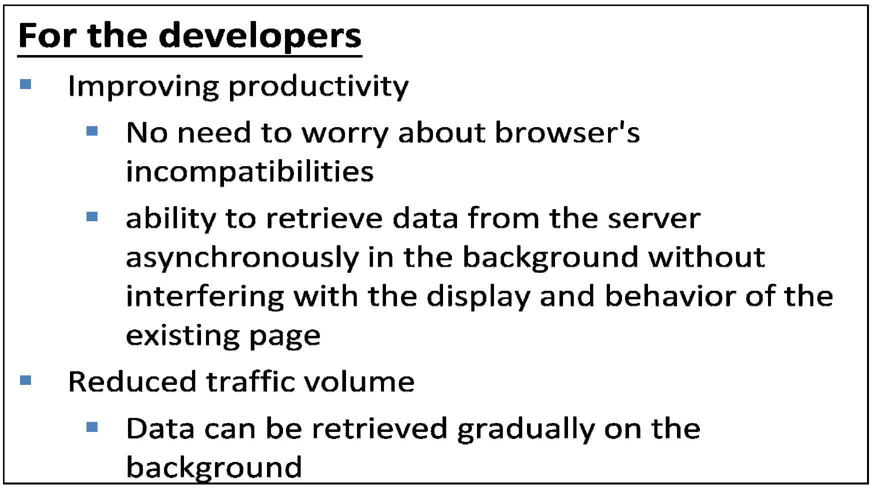
How are web applications different from traditional desktop applications?





At the client's end

- Support richer UI and interaction
  - Drag and drop
  - Video and audio facilities
  - Animation



For the developers

- Improving productivity
  - No need to worry about browser's incompatibilities
  - ability to retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page
- Reduced traffic volume
  - Data can be retrieved gradually on the background

RIA Characteristics

A number of key features differentiate RIAs from traditional Web applications.

Direct interaction: In a traditional page-based Web application, interaction is limited to a small group of standard controls: checkboxes, radio buttons and form fields. This severely hampers the creation of usable and engaging applications. An RIA can use a wider range of controls that allow greater efficiency and enhance the user experience. In RIAs, for example, users can interact directly with page elements through editing or drag-and-drop tools. They can also do things like pan across a map or other image.

Partial-page updating: Standard HTML-based Web pages are loaded once. If you update something on a page, the change must be sent back to the server, which makes the changes and then resends the entire page. There's no other way to do it with HTTP and HTML. With traditional Web-based apps, network connectivity issues, processing limitations and other problems require users to wait while the entire page reloads. Even with broadband connections, wait times can be long and disruptive.

But RIAs incorporate additional technologies, such as real-time streaming, high-performance client-side virtual machines, and local caching mechanisms that reduce latency (wait times) and increase responsiveness. A number of commercial development tools (see below) permit this partial-page updating.

Rich Internet applications: The tools

The list of current technologies that can be used to build modern RIAs is long. Here are some of them:

- AJAX
- Adobe Flash, Flex and Adobe Integrated Runtime (AIR)
- Microsoft Silverlight
- Curl (an object-oriented language with embedded HTML markup)
- Google Gears
- OpenLaszlo and Webtop
- Oracle WebCenter

Better feedback: Because of their ability to change parts of pages without reloading, RIAs can provide the user with fast and accurate feedback, real-time confirmation of actions and choices, and informative and detailed error messages.

Consistency of look and feel: With RIA tools, the user interface and experience with different browsers and operating systems can be more carefully controlled and made consistent.

Offline use: When connectivity is unavailable, it might still be possible to use an RIA if the app is designed to retain its state locally on the client machine. (Developments in Web standards have also made it possible for some traditional Web applications to do that.)

Performance impact: Depending on the application and network characteristics, RIAs can often perform better than traditional apps. In particular, applications that avoid round trips to the server by processing locally on the client are likely to be noticeably faster. Offloading such processing to the client machines can also improve server performance. The downside is that small, embedded and mobile devices -- which are increasingly common -- may not have the resources necessary to use such apps.