



Vinton G. Cerf

DOI:10.1145/2872420

Apps and the Web

Much has already been written about the differences and even the rivalry between native mobile and mobile Web applications. Part of this seems explainable owing to

the limited display and interaction capacity of a mobile. Specialized applications that use limited display space and soft keyboards have been crafted to match the typical mobile environment. Ironically, mobile apps and browser-based applications both rely on servers on the Internet for much of their functionality. Actions by users of either interface reach common ground at Internet servers and the results may be visible by either the Web or mobile operating system interfaces (consider email and social networking applications, for example). While there is standardization in the form of HTTP and various versions of HTML for Web-based applications that allow a browser to pull or push content from or to Web servers, interaction among mobile apps is rarer for lack of commonality other than sharing the interface to the mobile platform itself and shared access to common information provided by the servers.


There are trade-offs to be found between a native mode implementation of an application and a browser-based implementation. State information is often kept within a native mode application so it can still function to some degree in the absence of access to the Internet while a browser is typically dependent on state information that is maintained at the Web server with the browser acting largely as a display service. Ironically, early implementations such as the Netscape browser and all subsequent ones have provisions for storing state information in the form of cookies so servers need not to retain session state across intervals when the network

connection is broken or users abandon a session. The reality is both kinds of application implementations have local processing capacity, especially with the advent of Java, JavaScript, HTML5, and other high-level language interpreters. Web-based applications often have the property that users can move from platform to platform (mobile, to tablet, to laptop) transparently because critical state has been kept on the server. Email is a good example. On the other hand, an ereader application for a mobile may be attractive for precisely the reason the ebook is locally stored and can be read, regardless of the status of access to the Internet.

In theory, Web-based applications are more portable across mobile operating systems to the extent these systems are consistent about interpreting the high-level language programs. Native mode applications, compiled to run in particular operating system environments (for example, Apple's iOS or Google's Android), may prove to be more efficient but have to be crafted to fit the operating system application programming interfaces and services available. Dependency on persistent Internet access and/or on substantial data transport between a mobile and its Internet server can have economic and performance ramifications. If the mobile data rates are limited, if there are limits to the total (for example, monthly) amount of data transferred without penalty, or if connectivity is spotty, the resulting performance may be unsatisfactory. Battery life is another major consideration. Implementa-

tions of applications that are sparing of computing and data transfer requirements will be attractive to mobile users who do not want to run out of power in the middle of a busy day.

It is also true the question is not binary. It is possible to implement hybrid applications in which some code is native mode and some is HTML-based by concealing the native code in an HTML wrapper. I am not sure how common such implementations might be and would be very interested to hear from readers with implementation experience whether or not this is a common practice. Indeed, I am very interested to hear whether this question has become moot, owing to increased mobile capacity, higher speed, more reliable access to the Internet, and longer battery life.

As the implementation of IPv6 penetrates further into the Internet, mobiles will have the ability to implement end-to-end Internet connections with other mobiles, with servers, and with Internet-enabled appliances. This introduces the possibility of peer-to-peer interactions between mobiles. Would that change the equation with regard to native mode or mobile Web implementation? Mobiles are also becoming the user interface of choice for interacting with the Internet of Things and one begins to wonder about the roles of Bluetooth, LTE, and Wi-Fi in this context, but we will have to leave that topic for another column. 

Vinton G. Cerf is vice president and Chief Internet Evangelist at Google. He served as ACM president from 2012–2014.

Copyright held by author.

Copyright of Communications of the ACM is the property of Association for Computing Machinery and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.