



# Will the Web Vanish?

Charles Petrie • University of St. Gallen, Switzerland

It's sometimes argued that certain technologies become so pervasive that they "disappear." We no longer agonize over what kind of electricity to supply to our house, via what kind of wires, from what kind of dynamo, and which sockets to install. It's standardized and all we see are the appliances that consume the electricity.

Here, we're going to take a much narrower view of this question than another article<sup>1</sup> that starts with the premise that the Web is effectively its applications, thus making the Web invisible by fiat. This column will be much more of a backward look than a prediction.

Before we talk about "the Web" becoming so pervasive, ubiquitous, and standardized that it becomes invisible, we should be clear on what "it" is (apologies to Bill Clinton).

Fundamentally, for the Web's first versions, there were two protocols: HTTP for transactions and HTML for display. Browsers and servers compliant with these protocols implemented the World Wide Web. This is still true today, except that the protocols have become dramatically more complicated and allow much more functionality.

## Old School

When I first saw and started implementing web-pages back in 1993, you really only needed to know 10 HTML commands to get started: beginning the webpage, beginning the content, paragraph breaks, and line breaks, along with their partner end commands (except, strangely, for line breaks) would allow you to write a page. Another slightly complicated command would let you make links to other pages. That's five. The other five were defining a header, including images, setting fonts, bolding text, and centering text.

Then, you could also (if inclined) use another three commands to create basic tables. Essentially,

this work was simple and incremental. More complicated commands existed for menus and posting, using choices to a background server that would then return another page constructed according to user choices. After that came RSS that would let the user subscribe to changes in a website. And then, even though Tim Berners-Lee disparaged the term "Web 2.0" (see [http://en.wikiquote.org/wiki/Tim\\_Berners-Lee](http://en.wikiquote.org/wiki/Tim_Berners-Lee)) when it emerged around 2006,<sup>2</sup> it implemented the kind of interactivity that he intended for the original Web.

But now old school was out. You had to at least know Cascading Style Sheets (CSS) just to make nice displays, and you had to use sophisticated Web creation software. It was no longer practical to write HTML by hand using a text editor (yes, I did and still do that, and my simple pages will be forever displayable, including my slide presentations, but I digress so much now.)

## The Interactive Web Was Still the Web

*IEEE Internet Computing* was planned and developed in 1996, with the first issue published in 1997. As the first EIC, I asked in my first column: "Will the Internet turn out to be like the TV and movie industries, a broadcast media that affects many but in which only a few create? Or will it become an emancipating media, interactive and connective, that allows more voices to be heard and more people to collaborate in larger and more complex projects?"<sup>3</sup>

I told our readers the answer was up to you and you came through. We have a Web now that lets us not only find information but calculate and interact in ways that were simply not possible in the last century.

For a good while now we've had wikis, blogs (even though this magazine has declined to

use them), and social media. These extremely interactive media that allow readers to make comments and express themselves in other collaborative ways are still recognizably the Web: there are versions of HTML and HTTP still in use. There are other protocols, especially for streaming layered on top, but the Web is still as defined, if more complex. It's not going away any more than is electricity.

The applications are becoming more complex and interactive as people adapt to the possibilities. When Friendster and then MySpace came out, no one saw much use for them. Facebook did it right and now many of us spend hours on it every day, but it would have been hard to write down the problem we were solving when we were persuading someone to sponsor the work.

Today we have endorsements on LinkedIn. It's really hard to say why these are useful right now. But we can, without much strain, imagine a future in which virtual teams come together to accomplish short-term special tasks based upon trusted descriptions of capabilities.<sup>4</sup> The Web is more and more about functionality rather than display.

Still, it's hard to imagine that researching something on the Web using search engines will just go away. Page displays, along with the poor, will always be with us. The Web as HTTP, HTML, and displays is unlikely to disappear, especially as a platform for advertisements that seek to distract us.

### The Mobile Revolution

We actually might not think of webpages as webpages in the future because of mobile devices. Mobile devices aren't just computers with small touch screens. They run apps: native, Web, and hybrid. What are these and what's the difference, and do we need "the Web" anymore?

Let's think of a Web app as a small program that someone downloads into a special "sandbox" managed

by a browser. It runs, perhaps using Javascript or something similar, and then goes away when the tab is closed, perhaps leaving behind some cookies for later use. But the program is gone. A hybrid app is typically a native app that calls the browser that runs a Web app.

A native app is a small program that's installed on the computer. Just as the browser is the platform for the Web app, there's an app platform built into the mobile device computer's operating system. There are two main differences from Web apps.

One is that all browsers are expected to respect the basic Web protocol and language specifications so that the Web apps are inoperable across browsers. Native apps must be specially written for their OS, currently either Apple or Android. Further, whereas Web apps are discarded when no longer used, native app executable code remains on the computer until uninstalled, even if it's never used at all. In addition, there are usually a few mobile device functions, such as the accelerometer, that a native app can access that Web apps can't yet, but this isn't a fundamental distinction.<sup>5</sup>

Here's the thing: the apps now all look alike, especially when developers use HTML 5. The new Web apps will still run in a browser, but the user might not be aware of this because there are no browser controls showing. All of the control will be in the Web app. And now, the browser is being used as a special-purpose application: the user can't open a new tab and research something s/he just read about.

This is how the Web might fade from user consciousness. We now look at our monitors and take food from our refrigerators, and we don't think about the electrical infrastructure behind them. Apps go one step further. Imagine that the only electrical sockets in your house were the ones being used by certain appliances and you couldn't even see those sockets: you

only saw the appliances. That's what it's like for apps now.

### New School

Search tools, such as Google and IMDb, have become apps. You don't need a full browser to invoke them. You do it by icons managed by the OS. This is the new way.

Being old school, I'm not sure I like this future. I'll keep a browser running somewhere, just as I write this column in emacs (look it up.) And maybe other ADD people will always want to keep open multiple tabs reflecting their current mental threads. Moreover, I (and my ilk) will resist having our mental landscape restricted by the app icons available on a screen.

So, yes, though the Web will always be with us, I reluctantly agree that Web technology as evidenced by browser use will become invisible to most users because apps (and their unknown successors) that may or may not use Web technology are just so convenient. I suppose it's for the best, but the readers of this magazine will always be aware of the basic protocols that continue to be used in the invisible wiring of the Web.

### References

1. V.S. Pendyala, S.S.Y. Shim, and C. Bussler, "The Web That Dissipates beyond the World," *Computer*, vol. 48, no. 5, to be published May 2015.
2. T. Foremski, US and Polish Web 2.0 Companies Swap Notes at Stanford," *Silicon Valley Watcher*, blog, 2006; [www.siliconvalleywatcher.com/mt/archives/2006/11/us\\_and\\_polish\\_w.php](http://www.siliconvalleywatcher.com/mt/archives/2006/11/us_and_polish_w.php).
3. C. Petrie, "The New Engineering Computer," *IEEE Internet Computing*, vol. 1, no. 1, 1997, pp. 4-5; [www-cdr.stanford.edu/~petrie/online/v1i1-webword.html](http://www-cdr.stanford.edu/~petrie/online/v1i1-webword.html).
4. C. Petrie, "Plenty of Room Outside the Firm," *IEEE Internet Computing*, vol. 14, no. 1, 2010, pp 92-96; [www-cdr.stanford.edu/~petrie/online/peer2peer/vision2010.pdf](http://www-cdr.stanford.edu/~petrie/online/peer2peer/vision2010.pdf).

## Peering

---

5. R. Budi, *Mobile: Native Apps, Web Apps, and Hybrid Apps*, Nielsen Norman Group, 14 Sept. 2013; [www.nngroup.com/articles/mobile-native-apps/](http://www.nngroup.com/articles/mobile-native-apps/).


---

**Charles Petrie** teaches and coaches the topic of innovation in design thinking at the University of St. Gallen, Switzerland (<http://dthsg.com/dt-at-hsg/>). He retired as a senior research scientist from the Stanford

University Computer Science Department. His research topics are concurrent engineering, enterprise management, and collective work. Petrie has a PhD in computer science from the University of Texas at Austin. He was a founding member of the technical staff at the MCC AI Lab, the founding editor in chief of *IEEE Internet Computing*, and the founding chair of the Semantic Web Services Challenge. He also

manages the Black Rock City Municipal Airport 88NV. Contact him at [petrie@cdr.stanford.edu](mailto:petrie@cdr.stanford.edu).

---

 Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.