

Cutting-Edge or Well-Established Web Technology

Terry Yin

January 11, 2015

“The global programming community is fashion-driven to a degree that would embarrass haute couture designers from New York to Paris.”

(Fogus & Houser 2011, forwarding)

Web programming is a fast changing area, nobody in this industry can afford totally missing the cutting-edge techniques. This doesn't mean we all have to use the most cutting-edge techniques in all our products. This is not the fashion industry. But we need to learn the new ideas and take advantage of the new techniques. In today's Internet industry, this is a matter of survival.

The definition of cutting-edge technology is it's lack of consensus and not fully tested [Wikipedia \(2014\)](#). That definition already suggests that cutting-edge is something that should be avoided in serious products. So the remaining question is only in what way we can participate the cutting-edge techniques and where we should apply them.

A cutting-edge technique is being there for a certain reason. It solves some certain problems, with a certain cost, within a certain scope and brings certain opportunities. It doesn't make sense to deny or adopt any technique without fully aware of all these certain things in any serious product. We should not forget that cutting-edge technique has already got one inherited cost, lack of maturity.

W3C (The World Wide Web Consortium) has the Rule Of Least Power [Berners-Lee & Mendelsohn \(2006\)](#), which “suggests choosing the least powerful language suitable for a given purpose”. Because the less powerful the language is, the more one can do with the data stored in that language. For example, data stored in JavaScript is more likely to be understood by anybody in the future than the data stored in a Java Applet, because JavaScript is less powerful. Another example is content that is stored in HTML format is still comprehensible with simple text editor (without losing any information), while a PDF can only be opened by a PDF reader.

In general, I'm comfortable adopting any technology that follows the Rule Of Least Power. For example, the markdown language is in plain text format which is supposed to be readable in its original format without any extra rendering. And it's often translated to HTML before it's actually used. That means you can always decide to use HTML directly and get rid of the markdown language when you don't want it. Another example is RESTful API that exchange information between client and server with data in JSON format. The underlying technology is the least powerful and well-established (JSON over HTTP), therefore it brings less risk and less dependency.

I would be more careful with technology that depends on more powerful feature or standards. I would also be careful with technology that makes me depend it too much.

When we cannot apply the cutting-edge techniques that is related to us because of it's not mature enough to fulfil our need, I think we can still participate in some other ways. These would include experimenting in side projects, contribute in open source and actively follow the trends and joining the discussion.

References

Berners-Lee, T. & Mendelsohn, N. (2006), ‘The rule of least power’, *World Wide Web Consortium, TAG Finding* .

Fogus, M. & Houser, C. (2011), *The Joy of Clojure: Thinking the Clojure Way*, 1st edn, Manning Publications Co., Greenwich, CT, USA.

Wikipedia (2014), 'Bleeding edge technology — wikipedia, the free encyclopedia'. [Online; accessed 11-January-2015].

URL: http://en.wikipedia.org/w/index.php?title=Bleeding_edge_technology&oldid=637592179