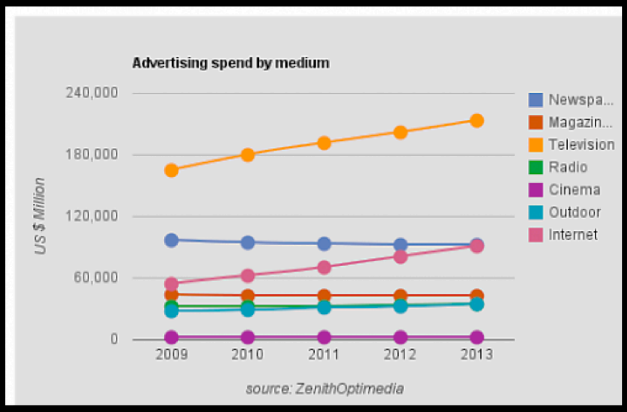
Hi there. Thanks for coming [blah, blah ☺]. Let me tell you a little about why Comcast just recently joined the W3C, and what our priorities are for this workshop and this group.

comcast

some context (*video*)*



http://paidcontent.co.uk/article/419-zenithoptimedia-all-media-bar-print-will-join-net-on-ad-growth-curve/

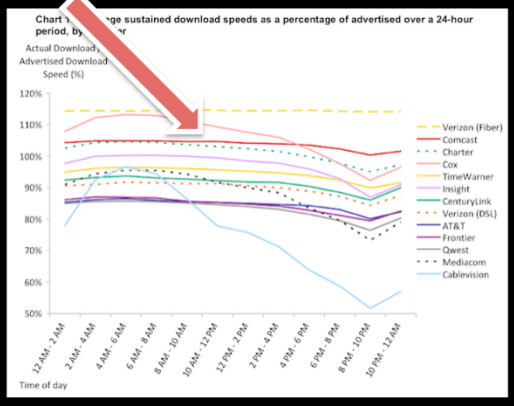
(*business is good – there are lots of metrics, but this is a good proxy)

First, some context: The video business is a good business. Lots of discussion about cord cutters, sub growth, sub loss, OTT, etc. Some good ways to evaluate objectively are to look at correlated. So, we're not here because the TV business is disappearing or turning a corner…

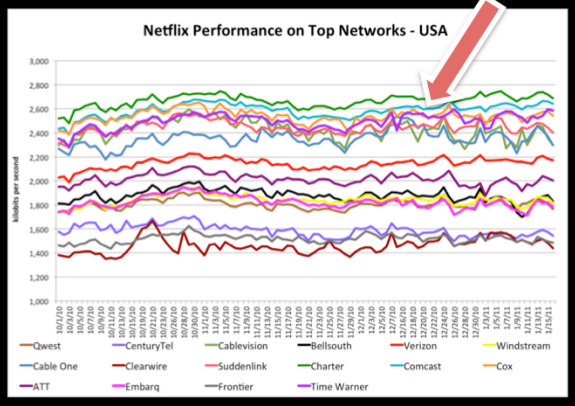
comcast

some context (*broadband*)*

comcast comcast



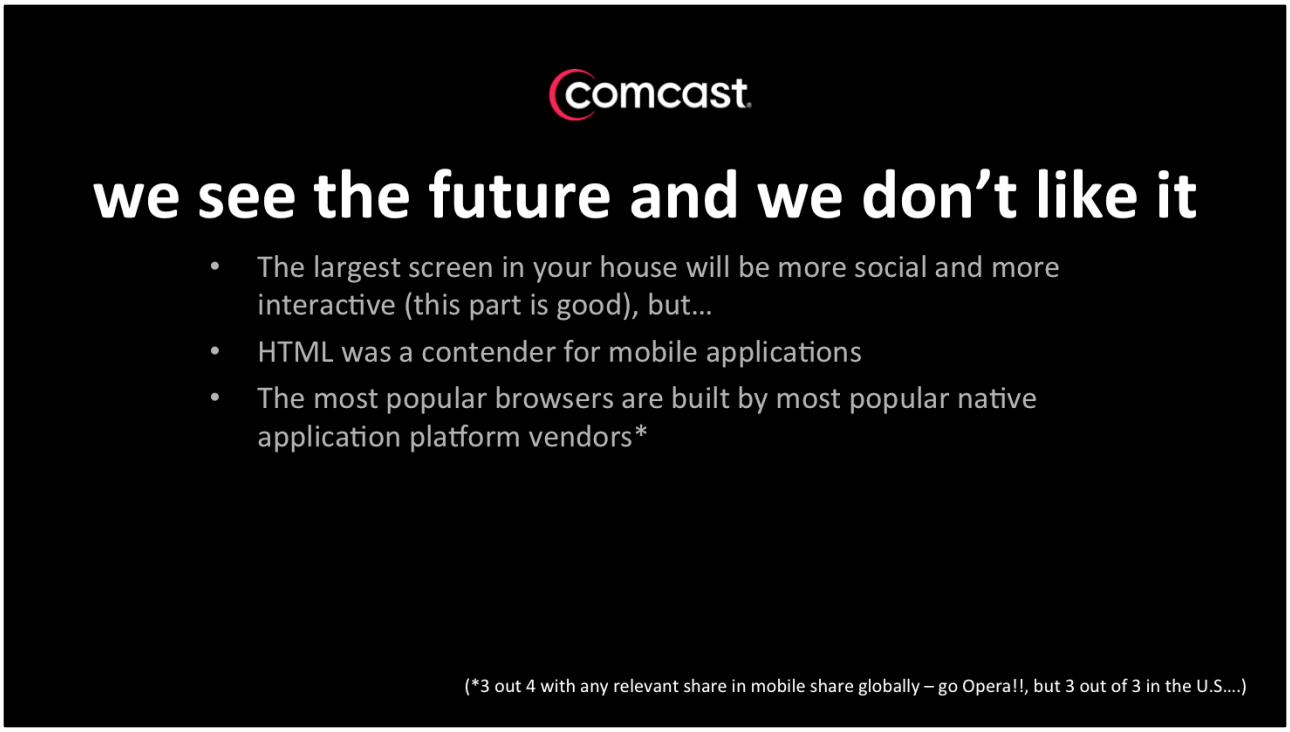
http://transition.fcc.gov/Daily_Releases/Daily_Business/2011/db0802/DOC-308828A1.pdf



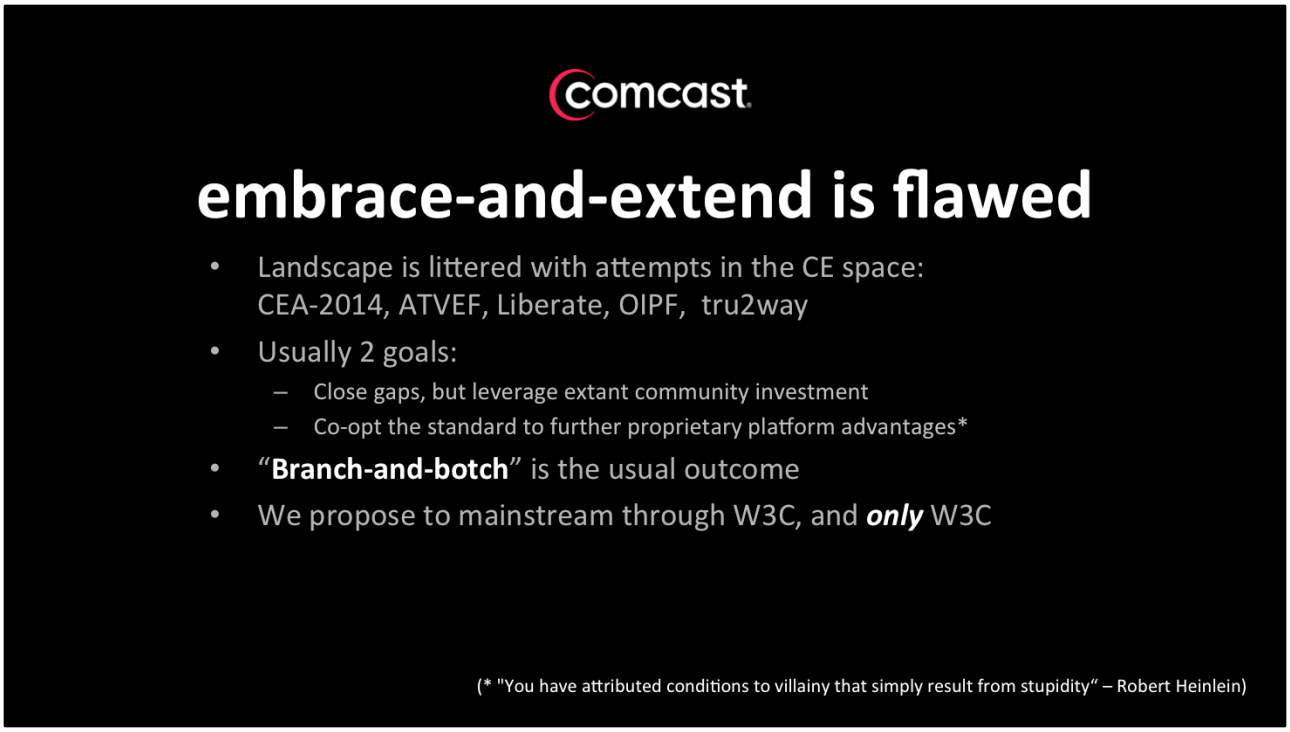
http://techblog.netflix.com/2011/01/netflix-performance-on-top-isp-networks.html

(*business is good – there are lots of metrics, but this is an example of "why?")

Second: The ISP/data/broadband business is a good business. Yes traffic growth is between 40-100% a year (depending on whose numbers you believe), and yes, it's surprising amount of real work, but again, we get good external confirmation that its working. We were the ONLY company to deliver better than 100% the speeds we promised, both upstream and downstream, in the FCC's report. And even Netflix reports we're amongst the top performing providers in the US for OTT video delivery. Customer speak with their wallets and their attention --- and we have both.



So what's our goal with the W3C? And why do we think HTML, specifically, is so important to our future state?
Media is going to be become more personalized, more interactive, and more engaging. Our role, as a value added, aggregator will necessitate incorporating and extending to new types of services, and our customers will therefore increasingly require a robust application platform. We could go it alone – we certainly have the scale of user base and consumer engagement to make it worthwhile. But we think that fragmentation in this space will impact the speed of service velocity, and the rate of innovation. [Imagine if MPEG didn't exist. Or if there were no aspect ratio requirements for video signals – if every piece of content had to be uniquely groomed for our service]. Finally, we've seen the failure in the mobile space. Increasingly, "endpoint" providers will not necessarily be the platform developers.

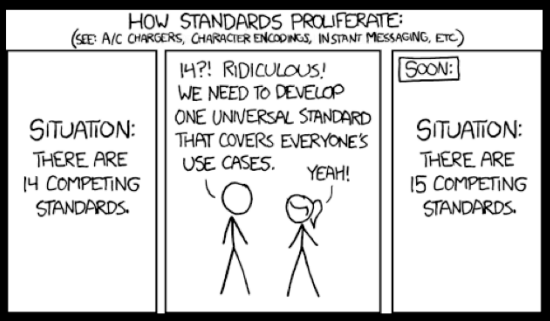


Another well-trod path, rather than build-your-own-from-scratch, is Embrace-and-Extend. Ultimately, we feel this is a tempting, but flawed strategy. Our goal is NOT lock-in, but healthy eco-system of content, media, and interactivity production.

That said, the reasons to build-your-own, or branch, are real and deep. They're not trivial problems. BUT, I don't think getting HTML to be "native"-like is the axiomatic goal. I think what we're trying to provide is hardened pathways to enable endpoint providers and content developers to solve the problems themselves. [go through points – for richness:lack of fast path, for "blindness": importance of home network]

comcast

core programmability issues



http://xkcd.com/927/

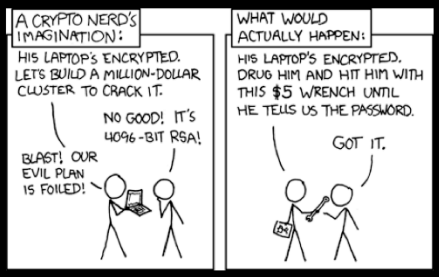
- A new standard is ***probably*** insufficient
- The video pipeline is opaque --- make it transparent and programmable (this is not just about triggers and events)
- Existence proof in Silverlight and Flash

Probably the core issue for Comcast is that the W3C standards around media, especially video, are leaning toward end-to-end solutions. The <video> tag, MPEG-DASH, etc. either provide opaque holes for browser vendors to dump in solutions, or, in the latter case, try to define pan industry standards – which we think, while a necessary attempt, will be slow to respond to rapidly changing market conditions. We think programmable video pipelines are the answer. The relative load is low, and there are excellent working examples today.

comcast

core security issues

- **Theft of service** requires proper access point authentication models
- **Theft of content** requires secure playback, but not necessarily "DRM"
- We believe these issues can be addressed by addressing security "tools", not (necessarily) providing an end-to-end solution (or standard)



http://xkcd.com/538/

This principle of programmability applies to security as well. "DRM" has been an ugly word in these circles, and for good reason, but providing a model for authentication of the software (both at app level and "engine level" that is more robust than user-agent is key. Similarly, to prevent theft of content, a secure mechanism for link layer content protection through to the device display is critical, and we think can be provided without being proprietary.

comcast

core composability issues

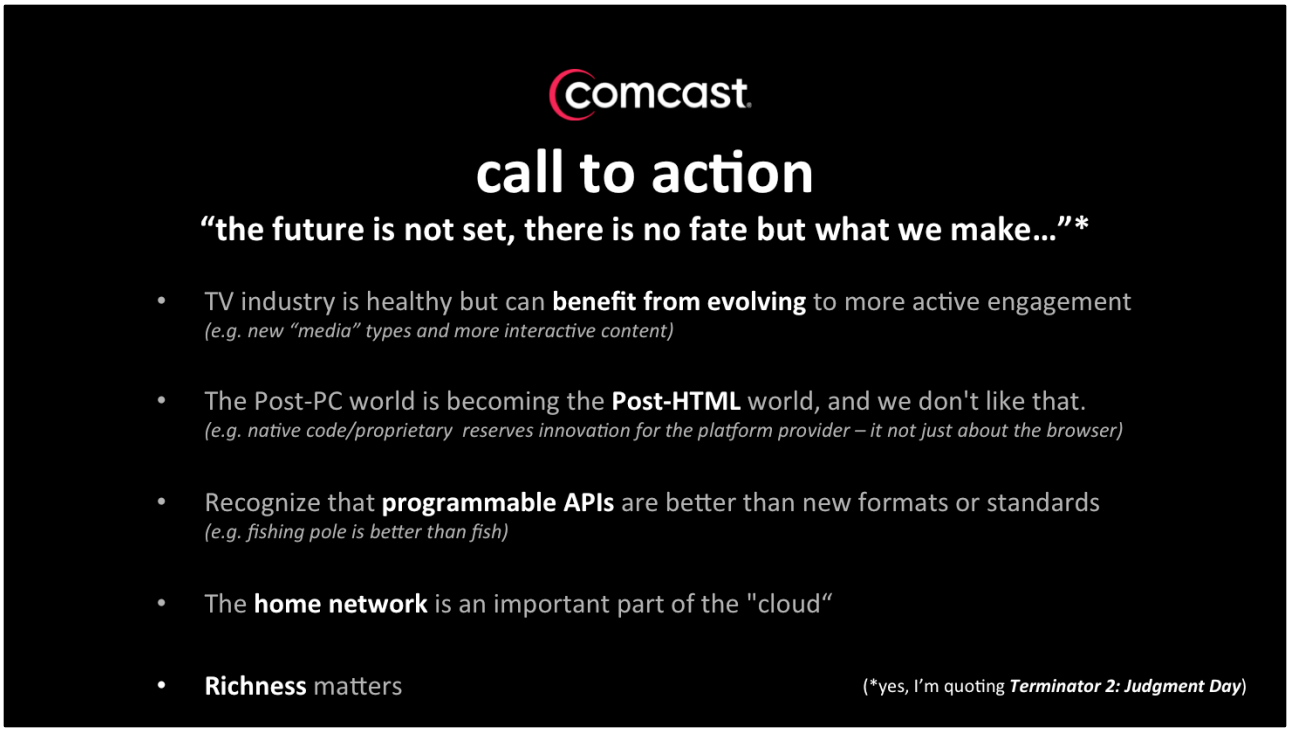
http://xkcd.com/198/

Imaginary use case: You want to write a browser in compliant HTML

- Can't securely embed applications
- Can't manage display or windowing
- Have to define my own packaging model (APIs, discovery, etc)

There are non-browser use cases for HTML – and they're becoming more important…

HTML has rightly been focused on the browser use cases as most critical. But a post-PC world is also looking to be increasingly a post-browser world. Turns out URL and forward/back is not the interaction paradigm that is carrying the day in the CE space. So these issues we're putting in front of you are not necessarily about the browser…



And there you go…