



Extending the Web: Maps, the Web Commons... and Pie


bkardell.com

The web..

← → ↻ 🔒 bkardell.com

🏠 ☆ 📧 🌐 📌 📅 21 🌈 📄 👤 ⋮

Brian Kardell



[Betterifying the Web](#)

- Developer Advocate at Igalia
- Original Co-author/Co-signer of The Extensible Web Manifesto
- Co-Founder/Chair, W3C Extensible Web CG
- Member, W3C (The JS Foundation)
- Co-author of HitchJS
- Blogger
- Art, Science & History Lover
- Standards Geek

Stuff about me...

I believe in the Web and its potential. I'd like to make it even better, and [I'd like your help](#).

My blog...

I'm in the process of making this domain the canonical source for all my posts. Here are a few of my favorites/some popular ones, but you can find a [complete list](#) (of the ones I've moved) or subscribe to the [rss feed](#) for updates.

- [Dropping the F-Bomb on Web Standards](#)
- [Performance: Shame, Hope, Nuance and Interesting Problems](#)
- [Potentially Scattered Thoughts on Web](#)

Websites for everyone!

- Social media
- Email
- Office Apps
- Graphical tools
- Online Banking
- Shopping
- Maps!
- etc..



Many desktop apps and mobile apps *also* use web engines

The screenshot displays a desktop environment with three overlapping windows. The background window is a web browser showing a document titled "ecosystem-health.md - ghostwriter". The document content is as follows:

Improving Web Ecosystem Health

For a lot of years, we've had lots of conversations and blog post on the topic of "browser engine diversity". I'd like to explain why I think this the wrong conversation and suggest a better way to frame the important parts: Ecosystem Health.

Back in January, Jeremy Keith compared the complexity of browser engine diversity to political systems...

If you have hundreds of different political parties, that's not ideal. But if you only have one political party, that's very bad indeed!

I like this analogy because I think he's right in even more ways than he intended. We don't even usually bother to talk about political parties in those terms because it's almost self-evident: 1 is too few, a hundred is too many - it's just chaos and noise. But what is the ideal number? This answer dogs us a lot, in part because it is not only unanswerable, it's actually kind of a red herring. I think there's something to this that leads to an important takeaway about how we think about the problem...

Numbers and goals...

The interesting part about this analogy is that the simple number of political parties isn't actually a very meaningful measure: Parties can differ a lot, or they can differ a little. 6 parties that barely disagree is quite a different thing than 3 that disagree substantially. Further, it's not simply a matter of giving voice to every dissenting opinion that matters either: There are political parties formed on ideas of hate. In short, what really matters is the *goal* of those numbers.

In the case of the political analogy, the real aim is about what a just model of governance. Similarly on the topic of "browser engine diversity", a real goal is "What makes for a healthy and effective ecosystem?" a talk about (and how) can easily lead us into perhaps the wrong take...

Older engines and diversity

It is fairly common that discussions of browser diversity point to examples of "why engine diversity matters": IE vs Netscape are oft able to dictate the market (or even kill it in this case) was a true, but very incomplete: There's a lot more beneath the surface.

Consider, for example, the fact that for a time there were *only* two very dominant ones (IE) was both entirely proprietary and based on a closed source.

Consider how different this could have been if IE been open source. Microsoft didn't want it to run on other operating systems, someone else. If Microsoft walked away from the Web, or worse, went belly over the project. If, for any reason at all, the primary maintainers of the project could more directly invest in advancements in the project.

A while later, Opera's Presto was very interesting, even multi-OS - but ultimately similarly proprietary.

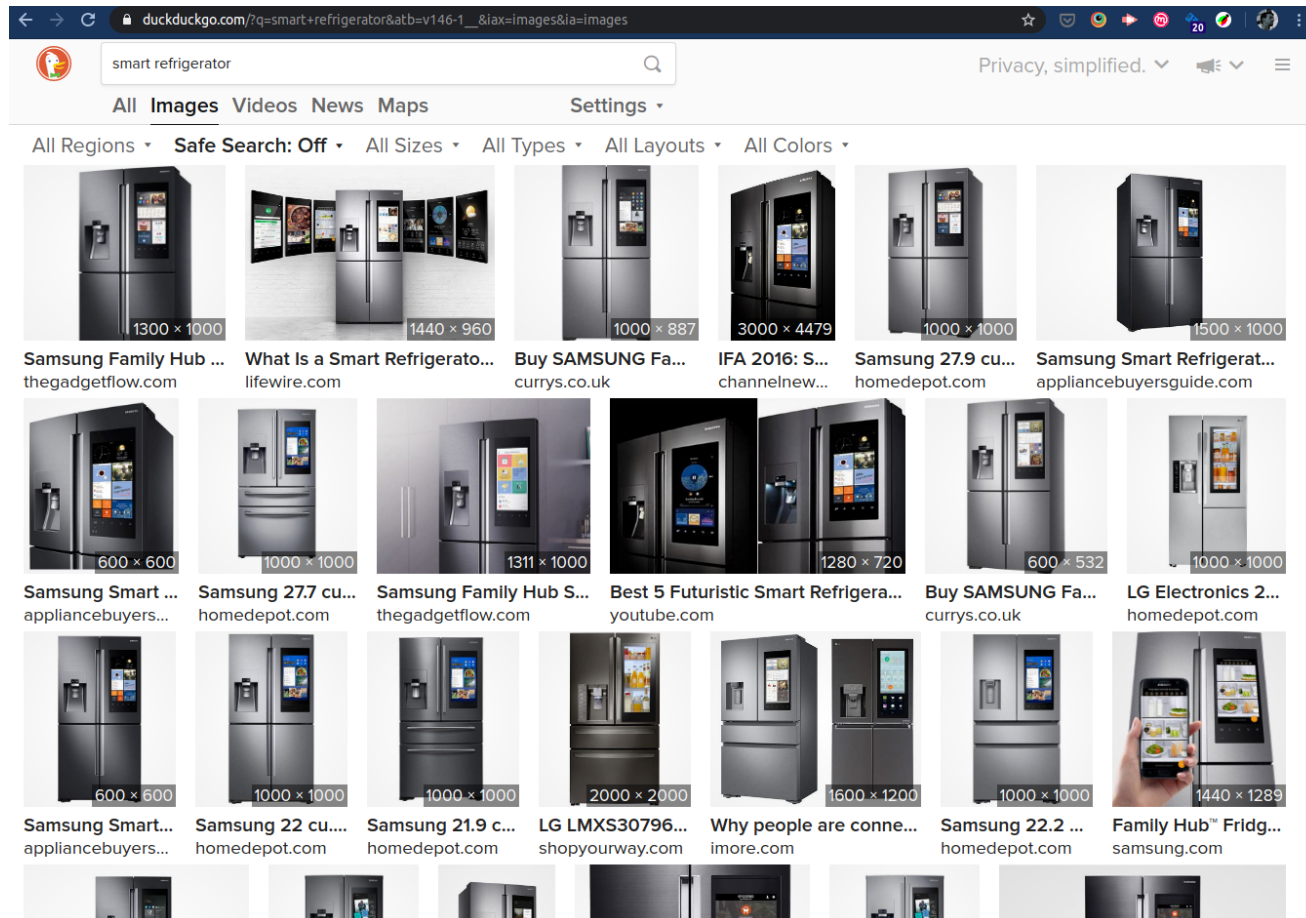
So, while we had greater 'diversity', these numbers alone aren't a great measure of the overall health of the ecosystem.

1,604 words

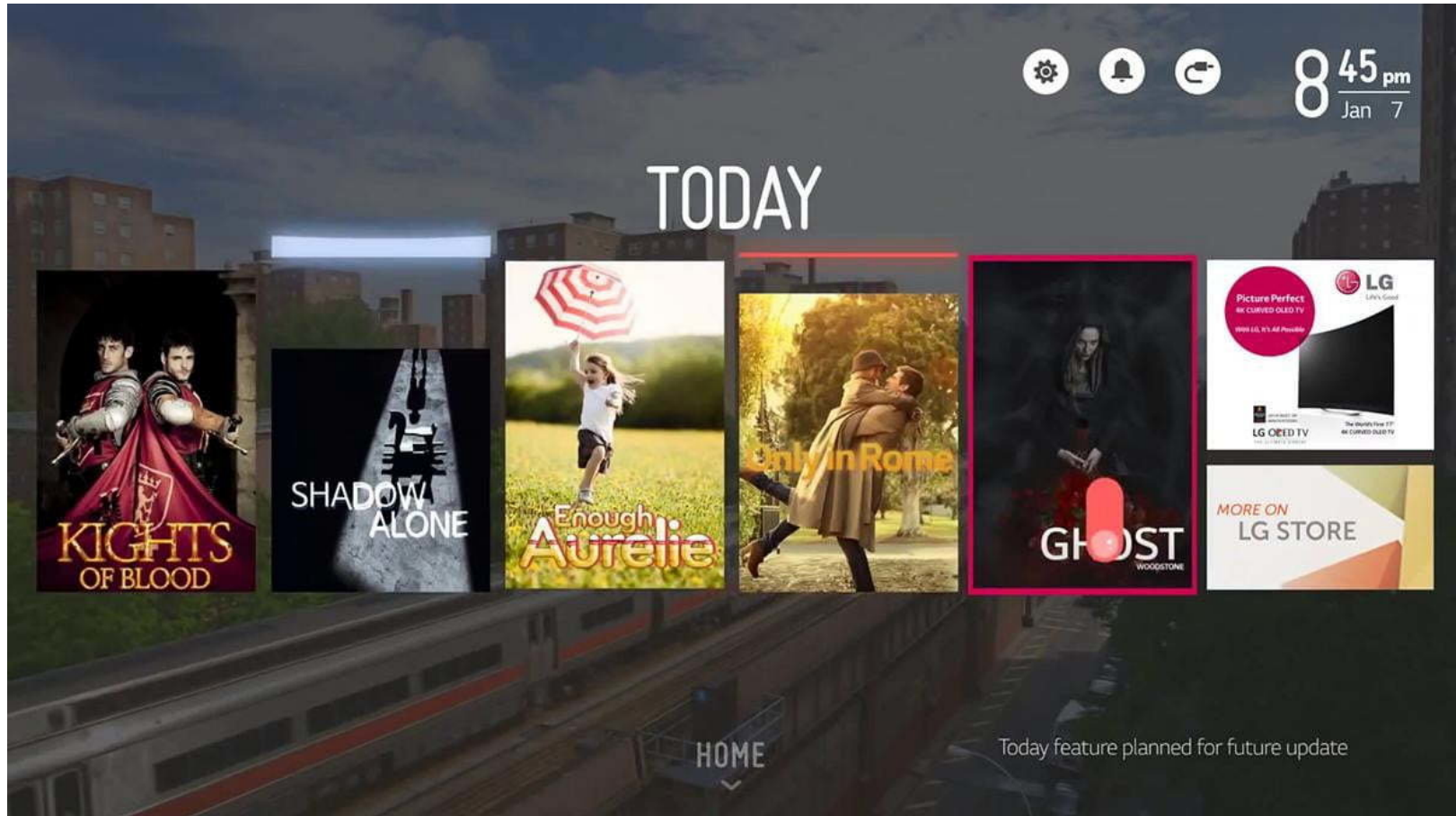
The foreground windows include:

- A Slack window titled "# Improving Web Ecosystem Health" showing a sidebar with various icons and a main area with the same text as the background window.
- A TaskOnBot window titled "TaskOnBot" showing a sidebar with "Threads", "Drafts", "Mentions & reactions", "Saved items", and "Channels". The main area shows a message from "TaskOnBot" with a "Next" button and an "End Tour" button.
- A Thread window titled "Thread" showing a message from "Brian Kardell" with the text "Hey @Michael Champion! So happy to see you here!" and a "Reply..." button.

Need a drink? There's a web engine!



Wanna watch TV? They use web engines!



Cars also use web engines



Digital signs... use web engines.



ding ding ding -- web engines!



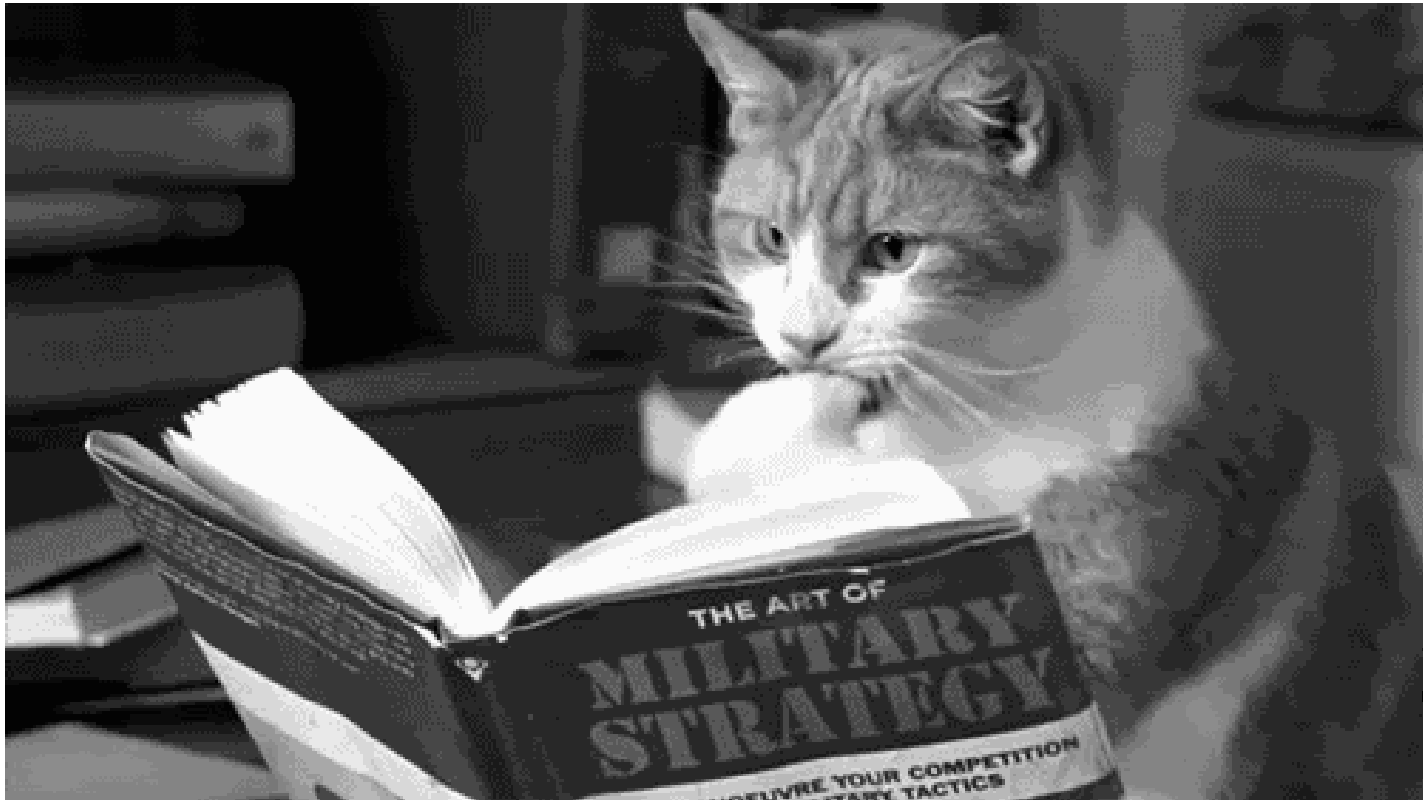
**As long as there's a
UI...**

You can't hide



No, really, you can't.

Many books are printed using web tech too.



The Web Platform is Infrastructure.. a commons.



But...

If you're a trying to improve it, that can feel... not good.

Your call is important to us. Please stay on the line until your cold, dead corpse turns to fossilized bone and archeological fragments.

somee cards
user card





The reason for this...

Pie



We're all prioritizing some time and energy, right now... is it worth this much of our pie?



Does everyone have the same budget? No.



I SPENT SO MUCH TIME AT THE DMV
THEY FINALLY JUST GAVE ME A LICENSE.

Some of us bring most of the pie.



But *importantly*:
Its still finite...
and voluntary.

It's not cheap.

Chromium: ~8,304 years of person effort

WebKit: 5,442 years of person effort

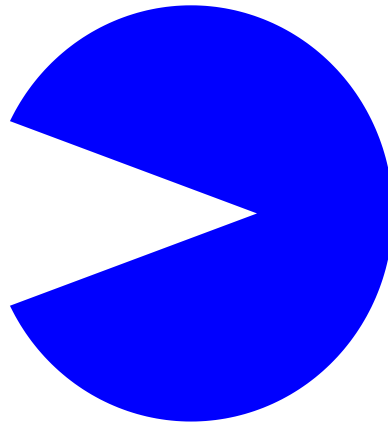
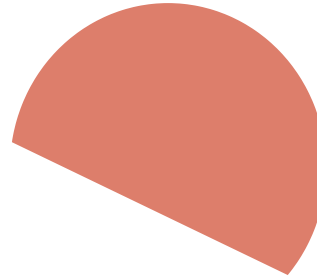
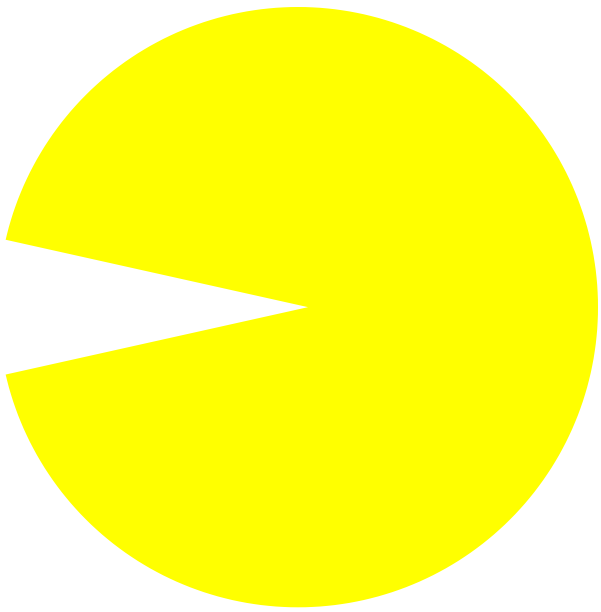
Firefox: 7,242 years of effort

(COCOMO model)



Wants are only accelerating





- Finite budgets
- Different sizes
- Different architectures
- Different constraints/priorities

Realistically, things *have* to be prioritized



Lots follows from this





**How do you
estimate this?**

Thus lots of stuff takes a long time to even get past here,
much never does (or hasn't yet)

30 years in:
~130 elements...
> 1/2 are 'spicy divs'

Tiny implementation cost, lots of standards costs

If you make it past that... this also can take a while

- agreeing there is even a thing is to discuss
- agreeing on details
- getting it done
 - ...everywhere
 - ...interoperability
 - ...accessibility kinks
 - ...styleability?
- education
- <developers are involved>
-still getting it wrong.

**Costs go up with
complexity...quickly.**

In 30 years: 1 non-form based interactive
standard UI control element

▼ Summary / Details

Very probably the simplest UI widget in existence. Very common in GUIs for almost as long as there have been GUIs.

Last major browser added support... last year.

... by way of attrition.

Then, *finally*, it's worth trying out for most people...

But they don't know that yet.

- agreeing there is even a thing is to discuss
- agreeing on details
- getting it done
 - ...everywhere
 - ...interoperability
 - ...accessibility kinks
 - ...styleability?
- education
- <developers are involved>
-still getting it wrong.

Then, way at the end...

- agreeing there is even a thing is to discuss
- agreeing on details
- getting it done
 - ...everywhere
 - ...interoperability
 - ...accessibility kinks
 - ...styleability?
- education
- <developers are involved>
-still getting it wrong.

How's that super expensive pie that took
a decade to bake taste?





https://extensiblewebmanifesto.org



The Extensible Web Manifesto

#extendthewebforward

We—**the undersigned**—want to change how web standards committees create and prioritize new features. We believe that this is critical to the long-term health of the web.

We aim to tighten the feedback loop between the editors of web standards and web developers.

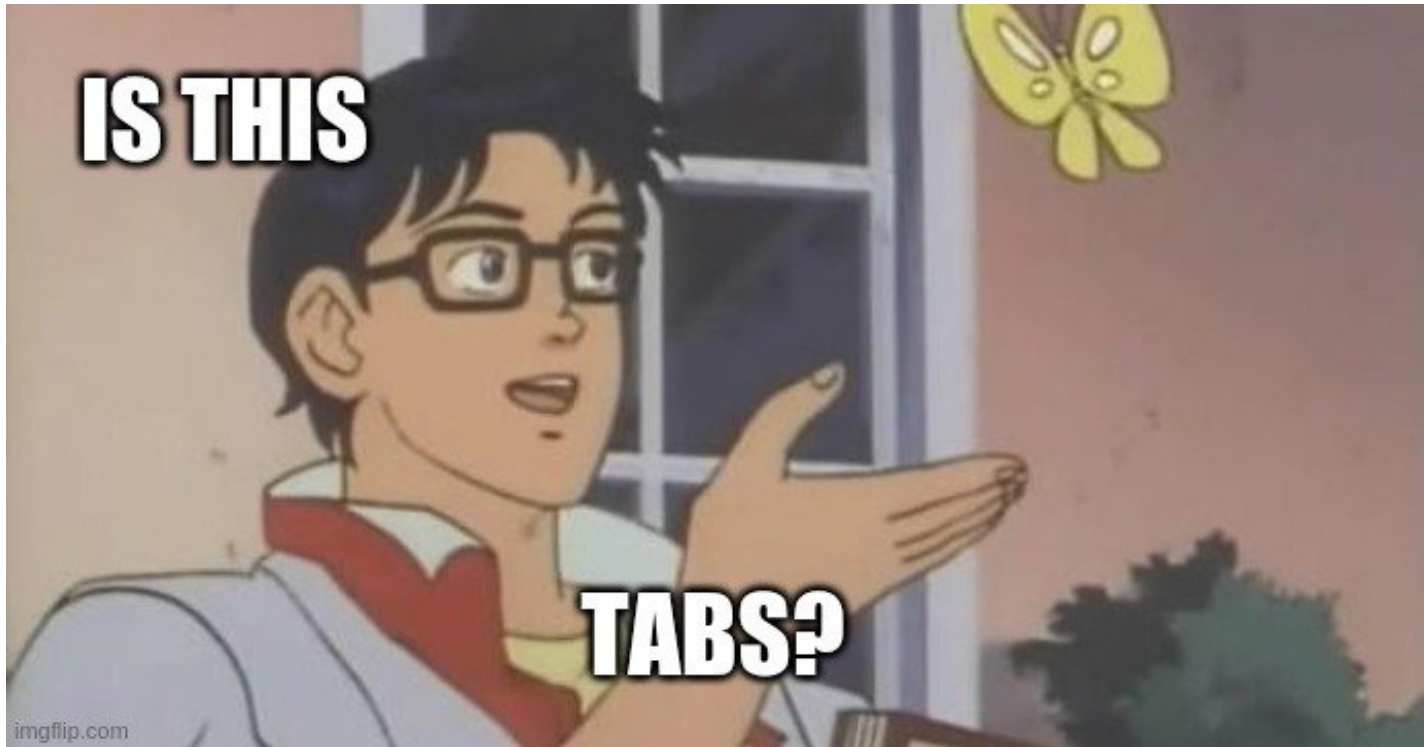
Today, most new features require months or years of standardization, followed by careful implementation by browser vendors, only then followed by developer feedback and iteration. We prefer to enable feature development and iteration in JavaScript, followed by implementation in browsers and standardization.

To enable libraries to do more, browser vendors should provide new low-level capabilities that expose the possibilities of the underlying platform as closely as possible.

They should also seed the discussion of high-level APIs through JavaScript implementations of new features (such as **Mozilla's X-Tags** and **Google's Polymer**).

IM JUST SAYIN'
YOU COULD DO
BETTER

The cow paths *aren't very clear...*



**Give developers mostly the
same abilities**

<x-details>

enter developers...



Do developers
actually like and use
it?

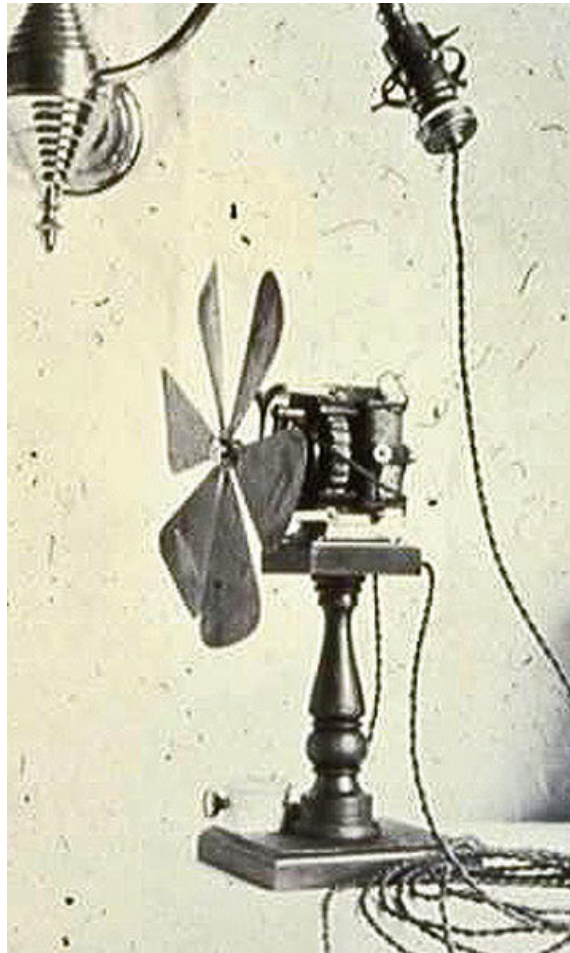




Failure is a feature.

It's necessary along the way.

Ingenuity: Adjacent Possibles





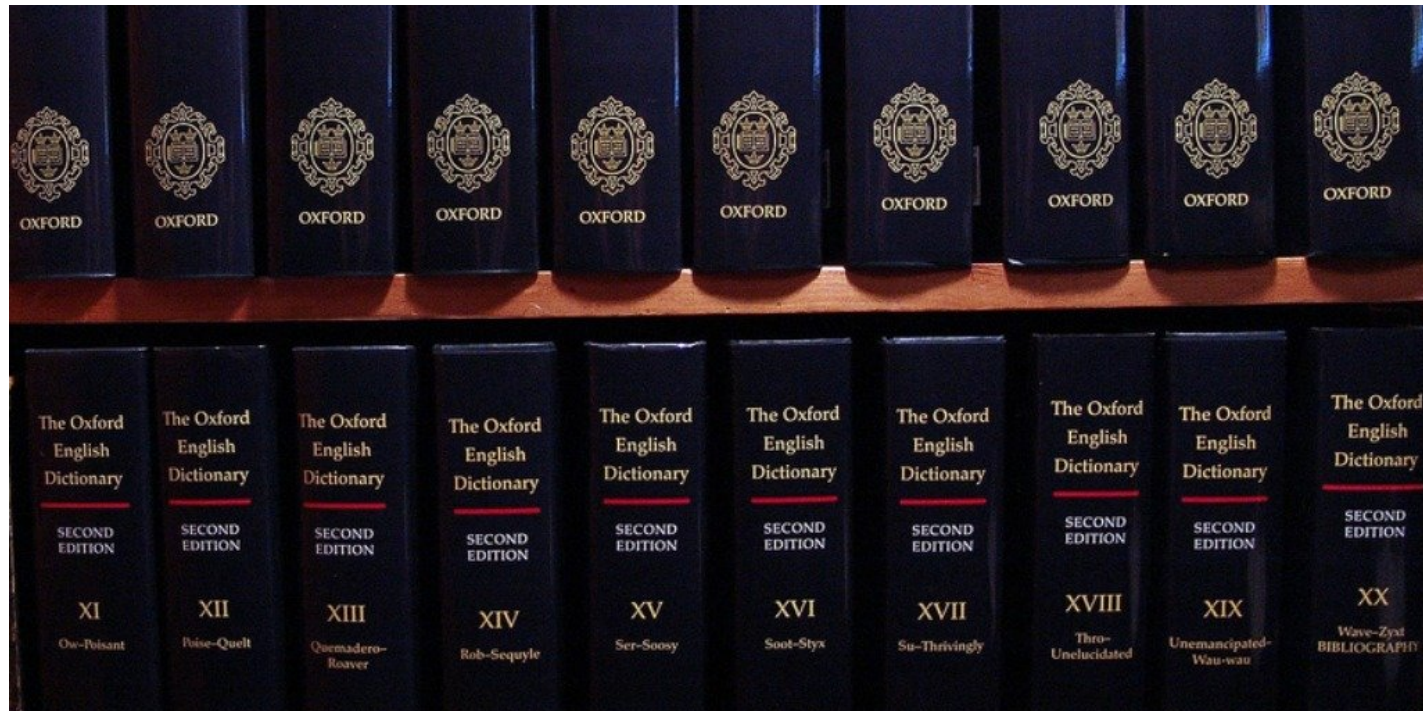
Many things

*wind up being fantastically successful at something they
weren't even designed for!*



Get out of the way.
Focus on enabling and science...

Write it down.

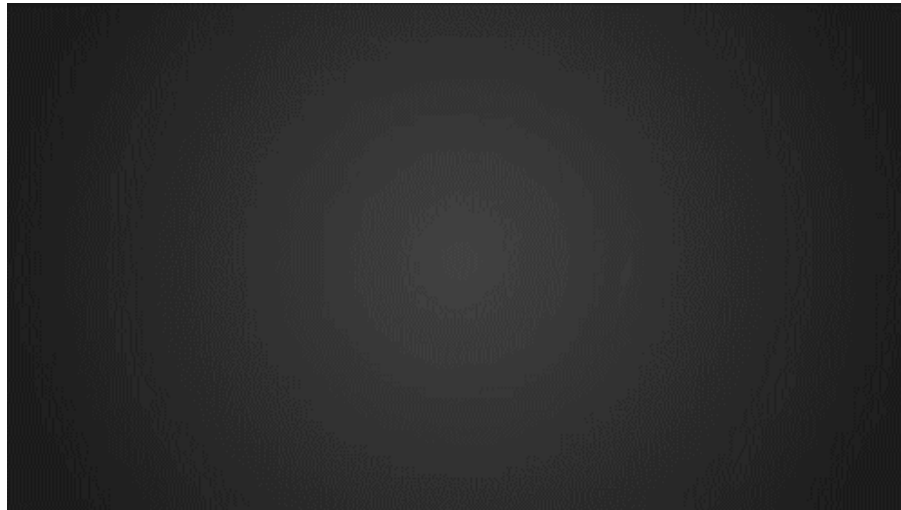


Important details



**A nice, bright,
unambiguous path
with proof and
estimability.**

...how does this map
well to the web
platform?



Break it down.

Reduce the magic required.

Expose the parts.

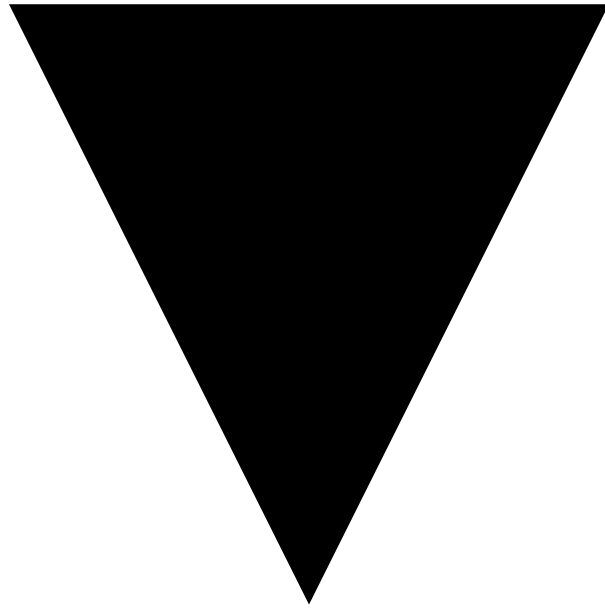
Aim for the middle.

<dialog>



</dialog>

<dialog>



inert:

a complex juggle of existing management of events,
attributes, text-selection, etc.

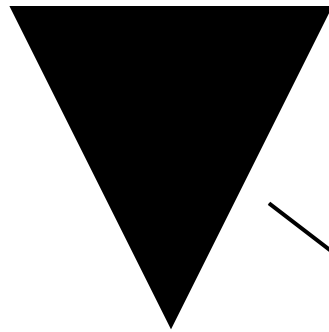
Necessary for dialog

but *also* useful for many *other* UI patterns which we also lack....

Including some 'common' ones that aren't remotely close to standardizable - like *drawers*



Pie required for *lots* of things



Pie required for dialog

Is this worth investment?



<x-drawer>?

- uses inert, like this



#extendthewebforward

resilience/adaptability



JSON vs XML

**Should the web have
maps?**

...yes?

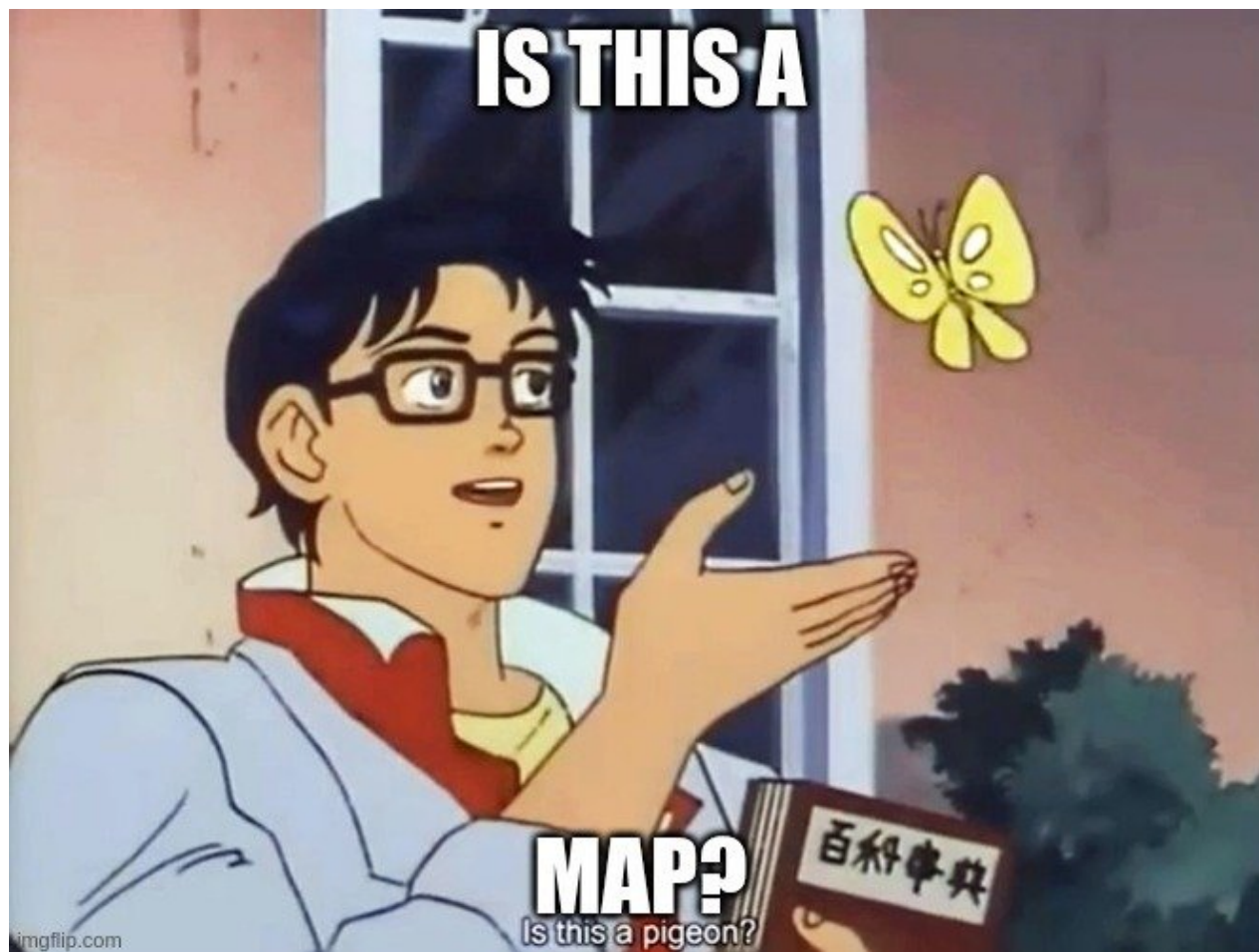
We need *so much more pie*



Just bring more pie.



**All of these things
apply...**



Maps: What is "good enough?"



Developers will know it when they see it.



But, it also has to *fit*
the platform well.

And is has to be aware of the pie.

Break it down.

Aim for the middle:

- Improve cost-benefit / find unlikely allies
- Lift all boats
- Make the higher-level proposals smaller with steps
- *Share the pie*



**What are the things
that make it hard
today?**

Are those things *only hard for you?*

Squint at it..

Very similar coarse grained needs



- An explorable map of Mordor for my LoTR site?
- Video game maps?

Even for things that are not maps at all?



- Product images for ecommerce?
- Technical Drawings?

Lacking features needed for (way more than) maps...

- Pan and Zoom?
- Offscreen Canvas?
- Hardware Accelerated SVG?
- Some larger partial/request rendering patterns?
- Probably a lot more?
- What are your ideas?!
- How can I help?

Thank you.