igalia

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

bkardell.com

The web..

bkardell.com

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...

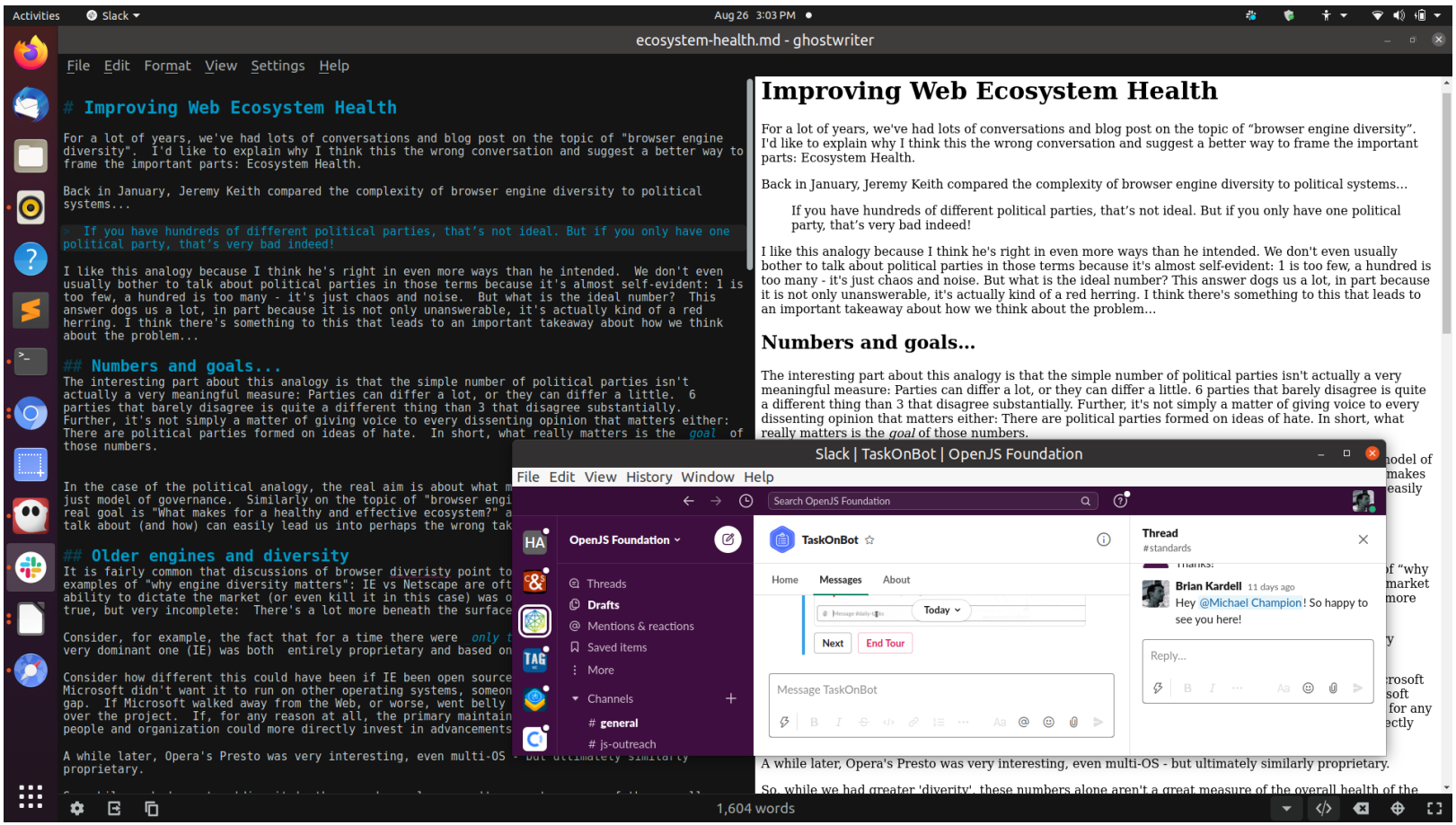
I'm in the process of making this domain the cannonical source for all my posts. Here are a few of my favorites/some popular ones, but you can find a complete list (of the ones I've moved) or subscribe to the rss feed for updates.

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

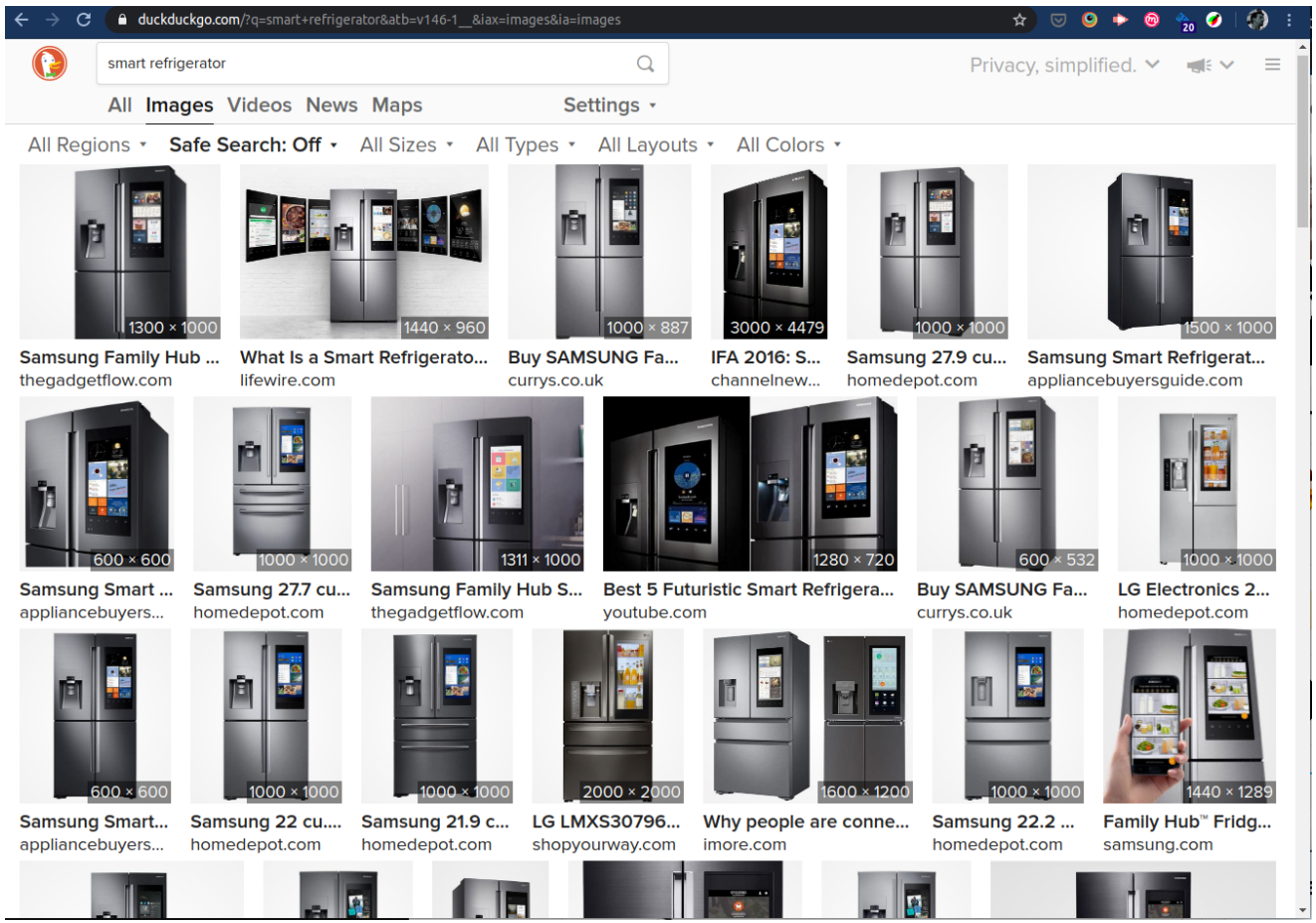
Websites for everyone!

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

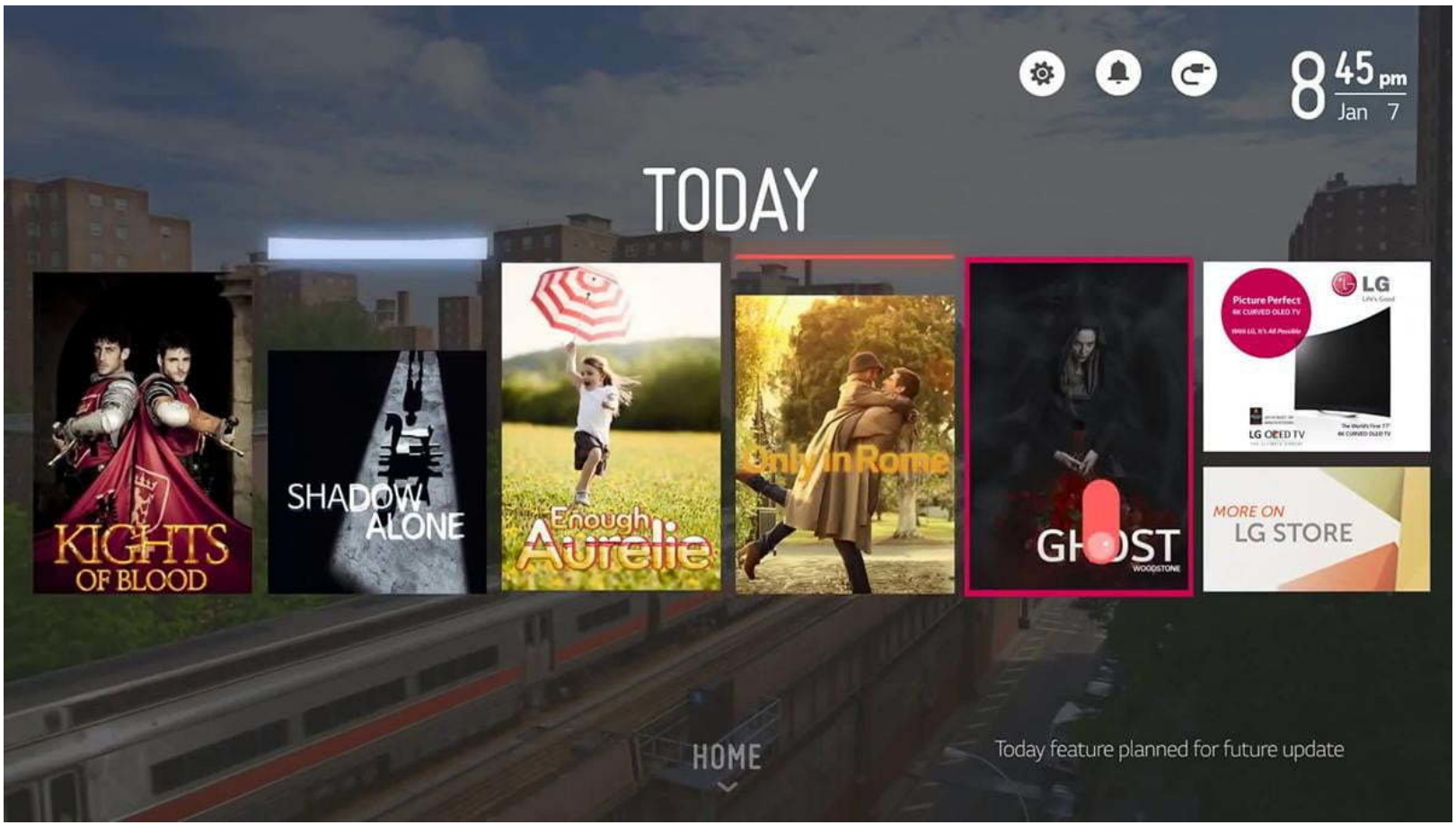
Many desktop apps and mobile apps *also* use web engines



Need a drink? There's a web engine!



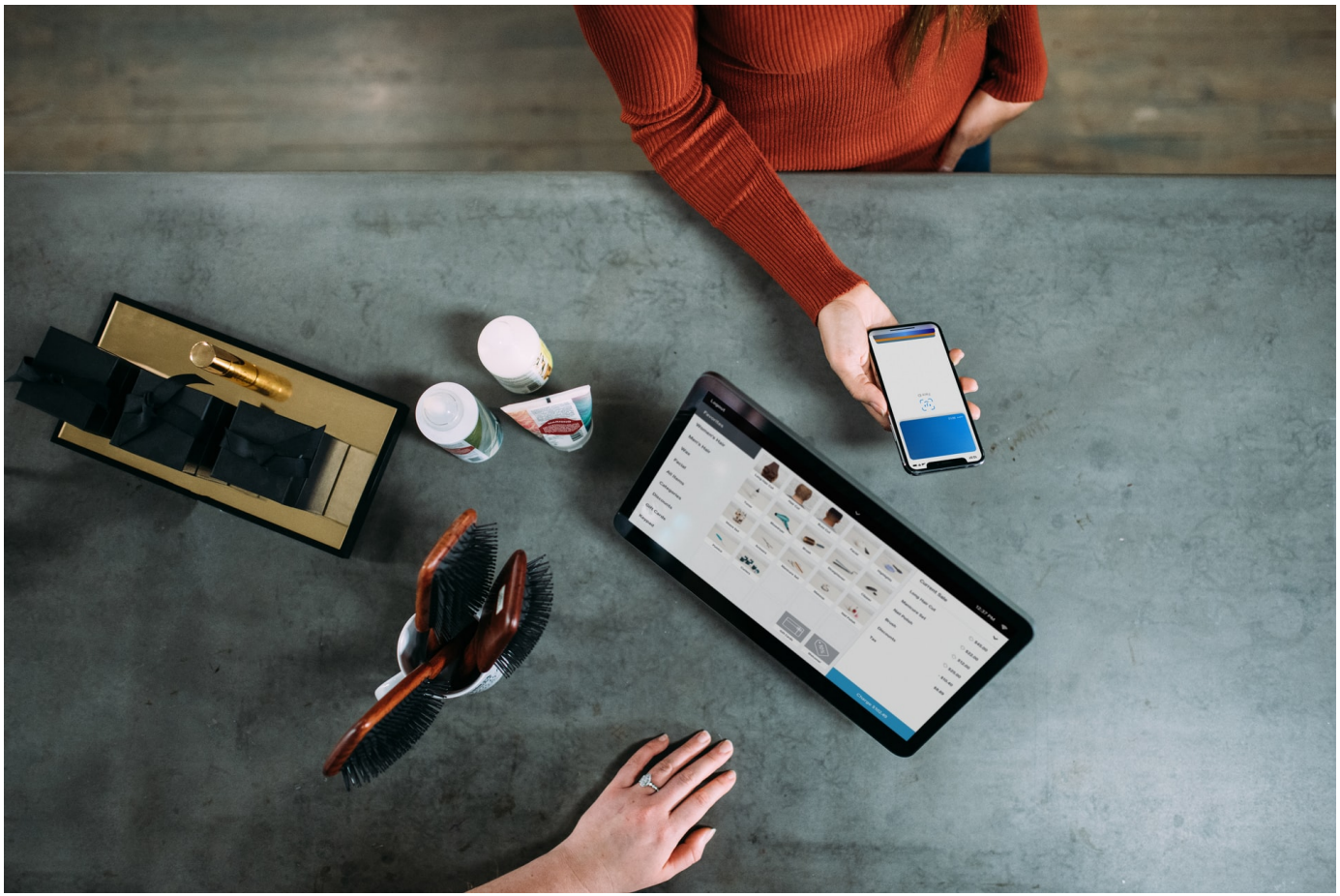
Wanna watch TV? They use web engines!



Cars also use web engines

Digital signs... use web engines.

ding ding ding -- web engines!

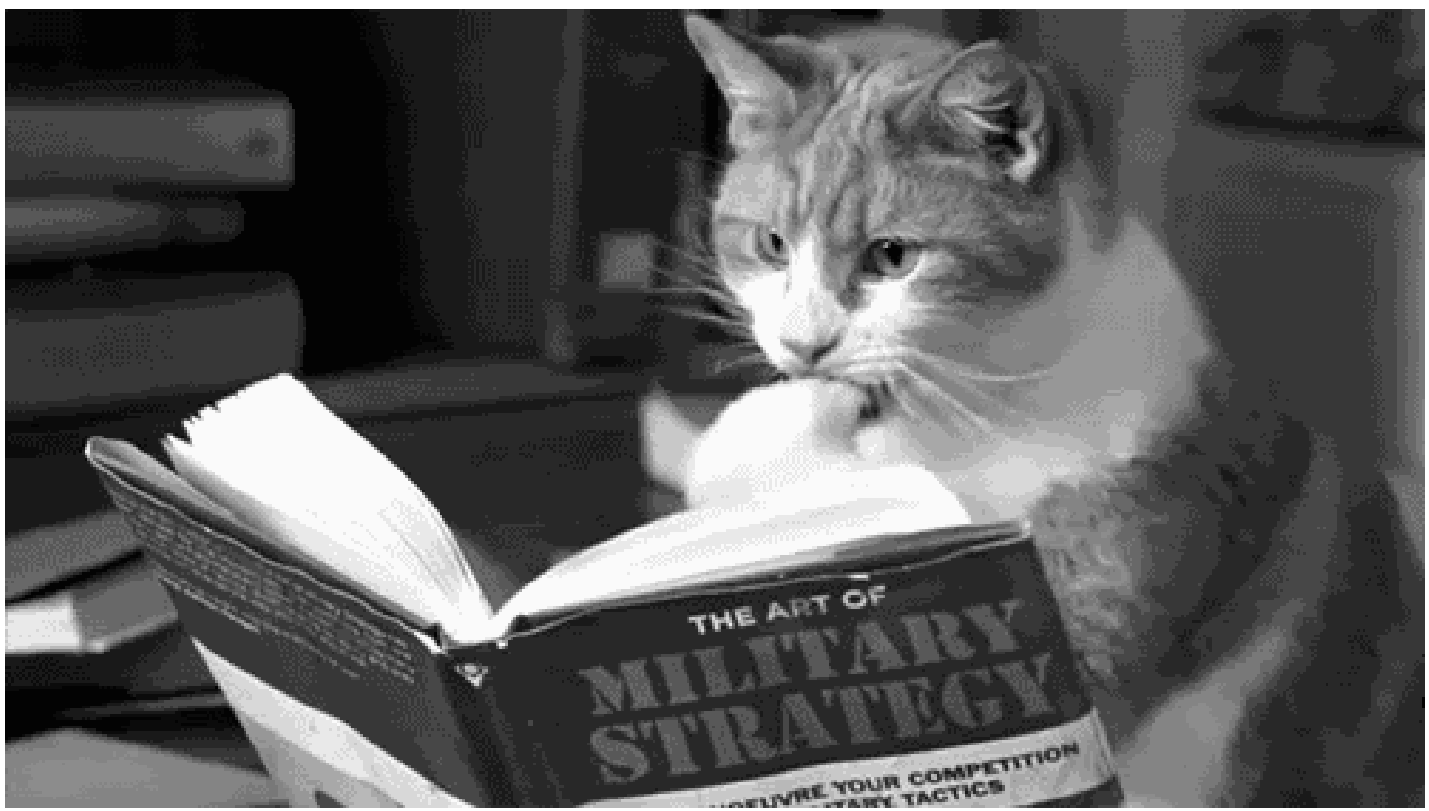


As long as there's a UI...

You can't hide

No, really, you can't.

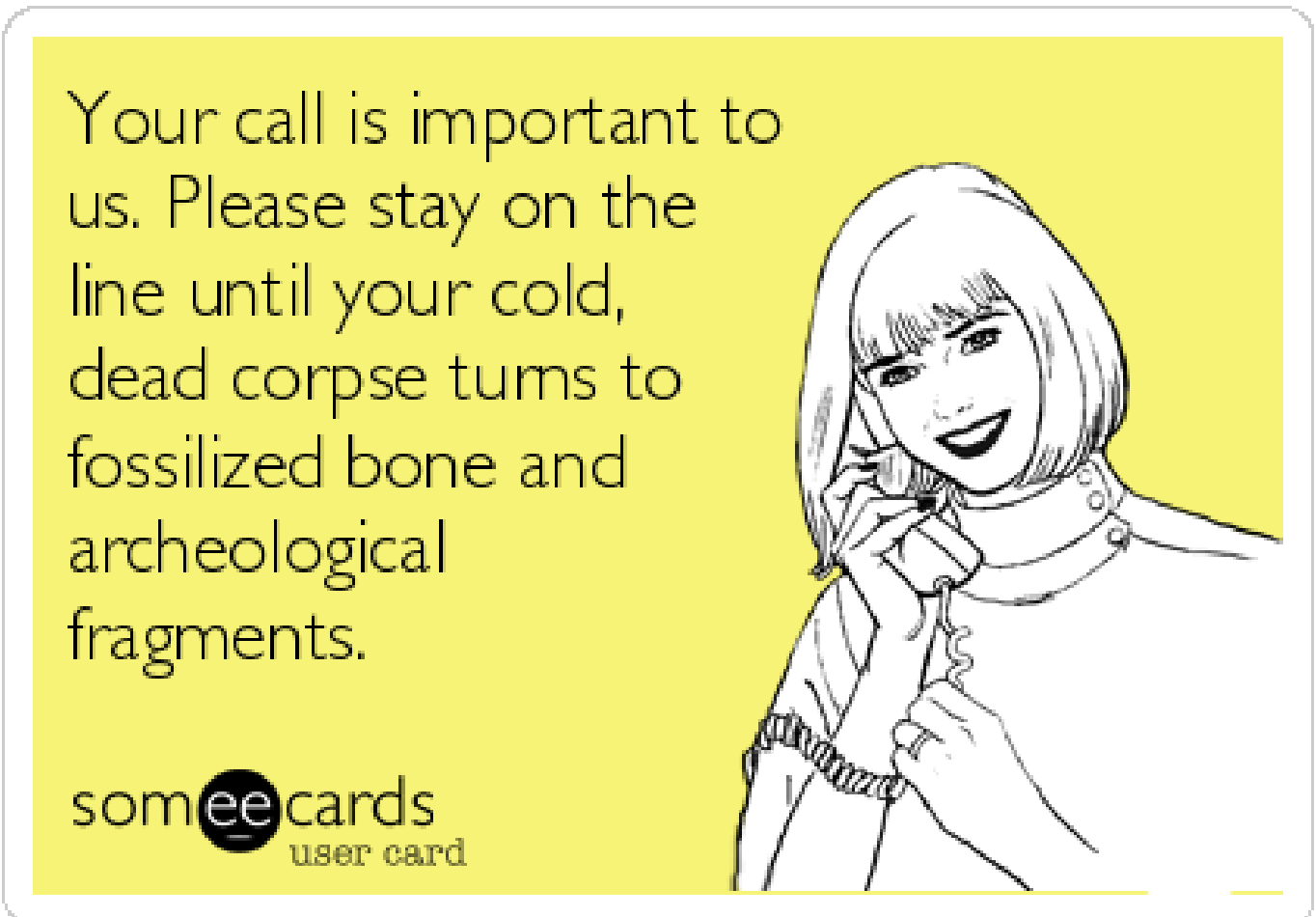
Many books are printed using web tech too.



The Web Platform is Infrastructure.. a commons.

But...

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



BILL NYE
SAVES THE WORLD
NETFLIX

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.

Some of us bring most of the pie.

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

It's not cheap.

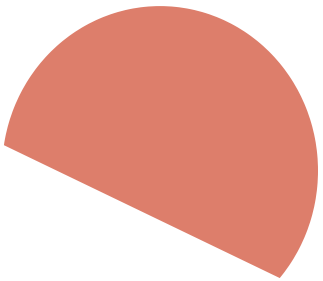
Chromium: ~8,304 years of person effort

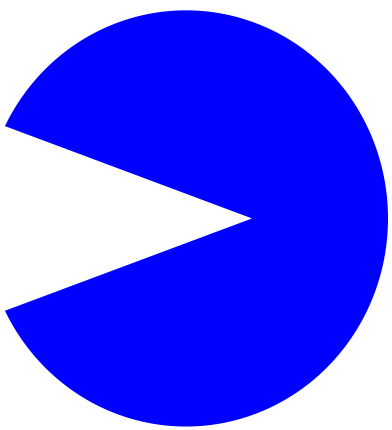
WebKit: 5,442 years of person effort

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.

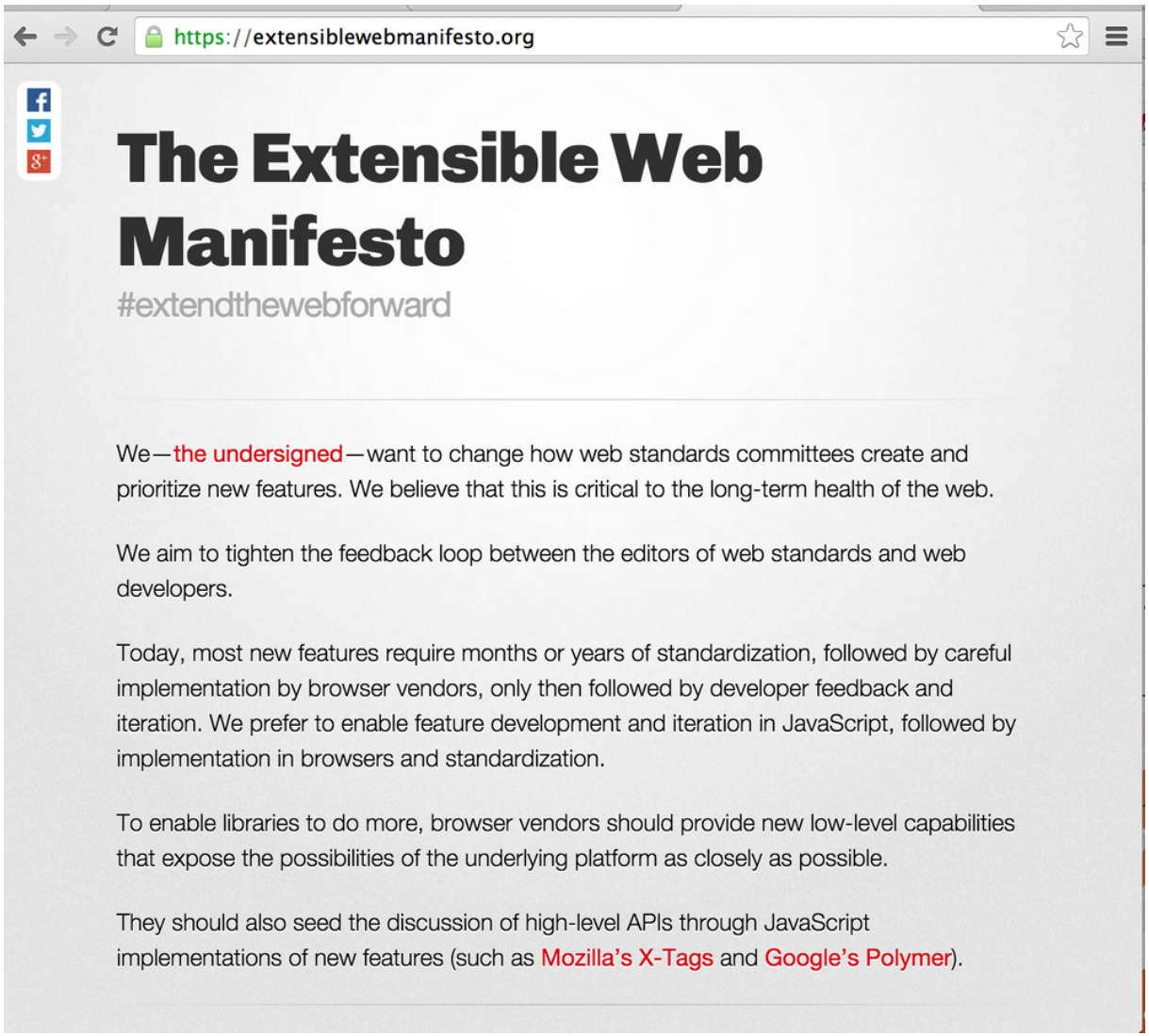
Then, *finally*, it's worth trying out for most people...
But they don't know that yet.

- 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...

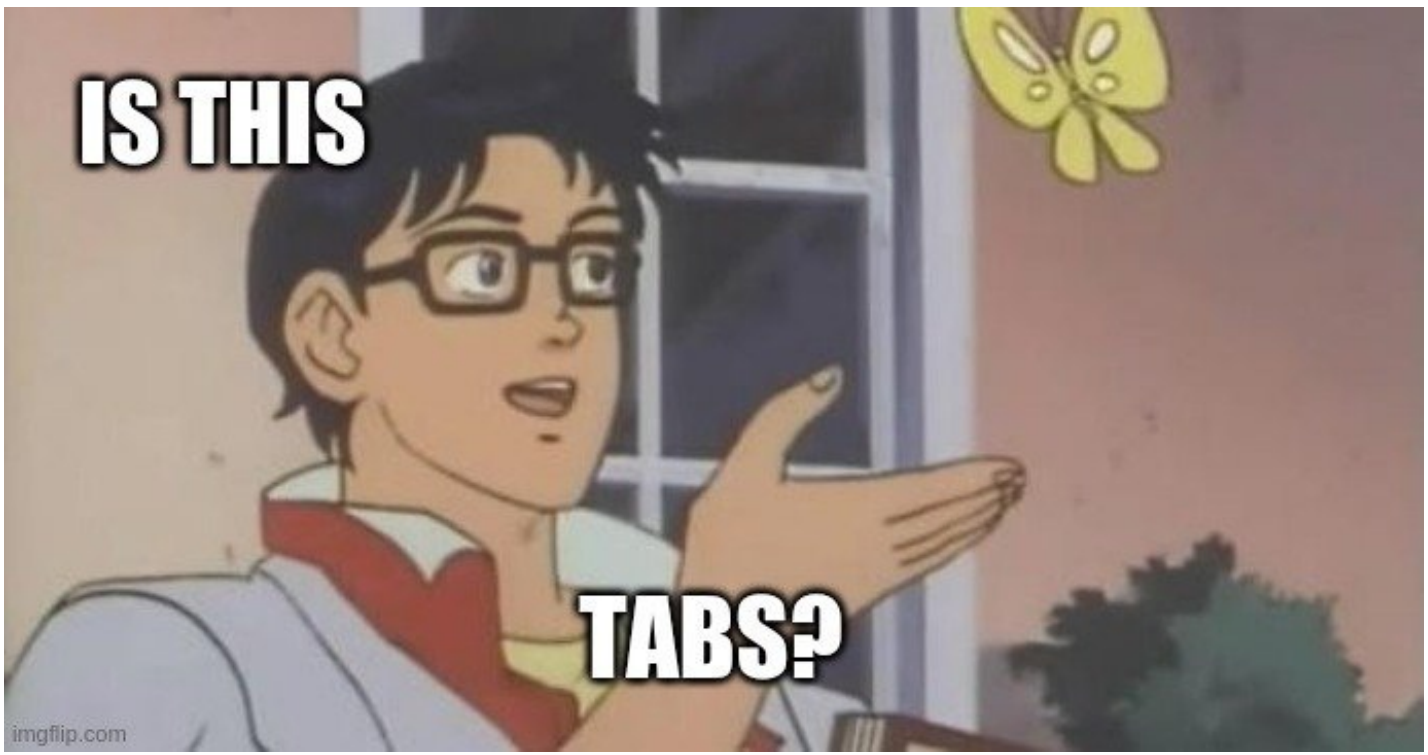
- agreeing there is even a thing is to discuss
- agreeing on details
- getting it done
  - ...everywhere
  - ...interoperability
  - ...accessibility kinks
  - ...styleability?
- education
- <developers are involved>
- ....still getting it wrong.

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).

YUFFIE.TUMBLR.COM

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



Give developers mostly the same abilities

<x-details>

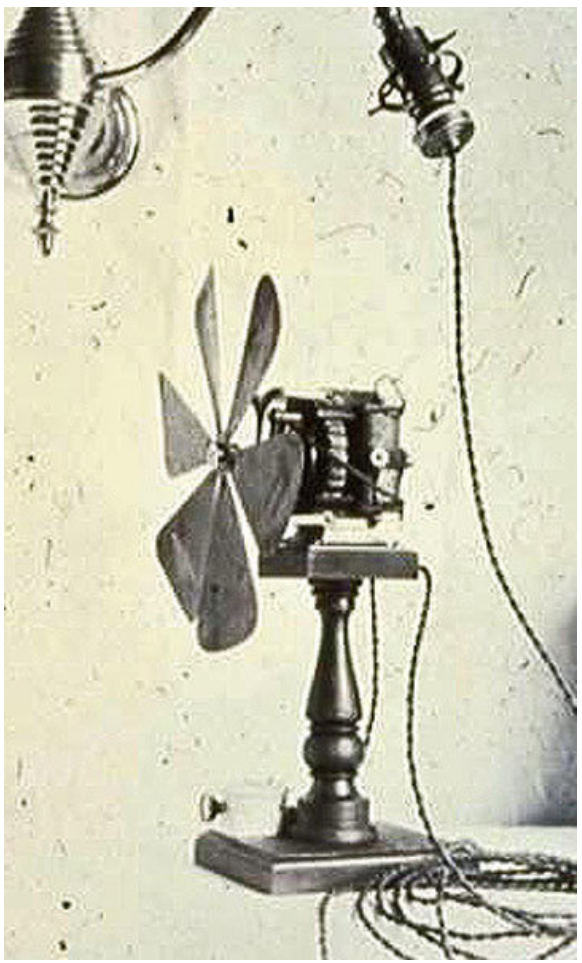
enter developers...

Do developers *actually* like and use it?

Failure is a feature.

It's necessary along the way.

Ingenuity: Adjacent Possibles



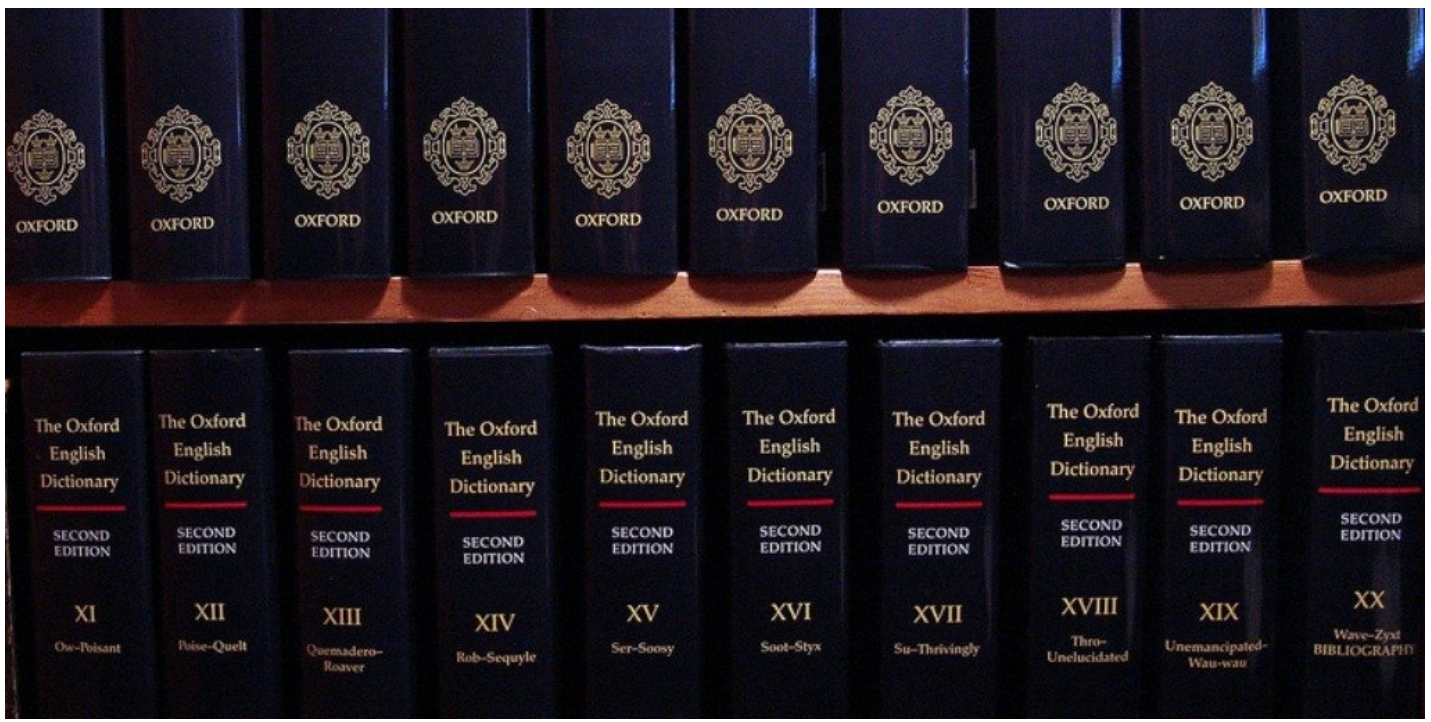
Many things

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

Get out of the way.

Focus on enabling and science...

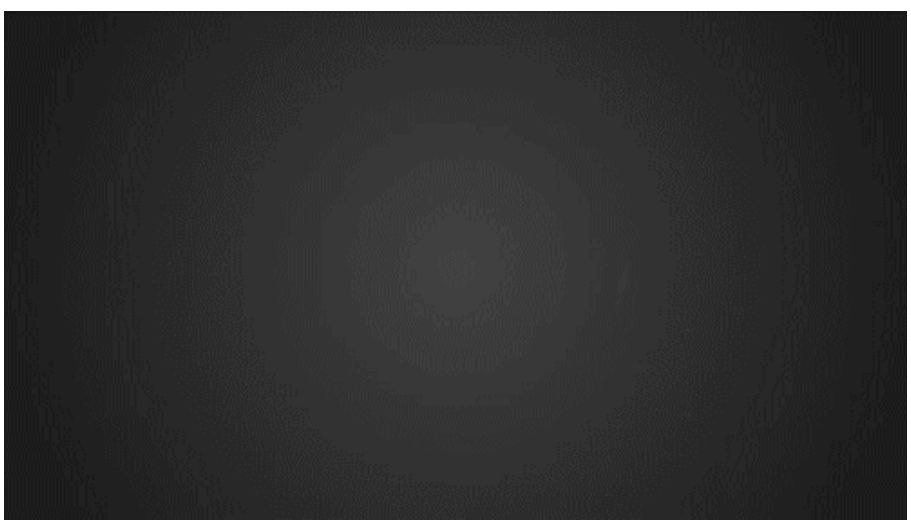
Write it down.



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.

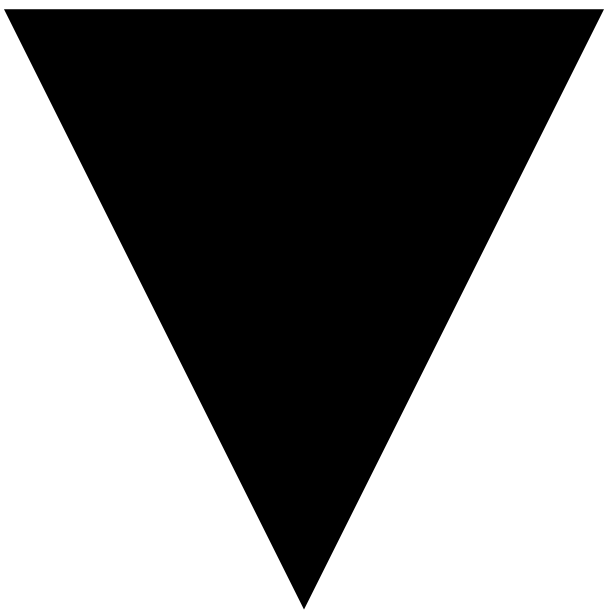
Expose the parts.

Aim for the middle.

<dialog>

</dialog>

<dialog>



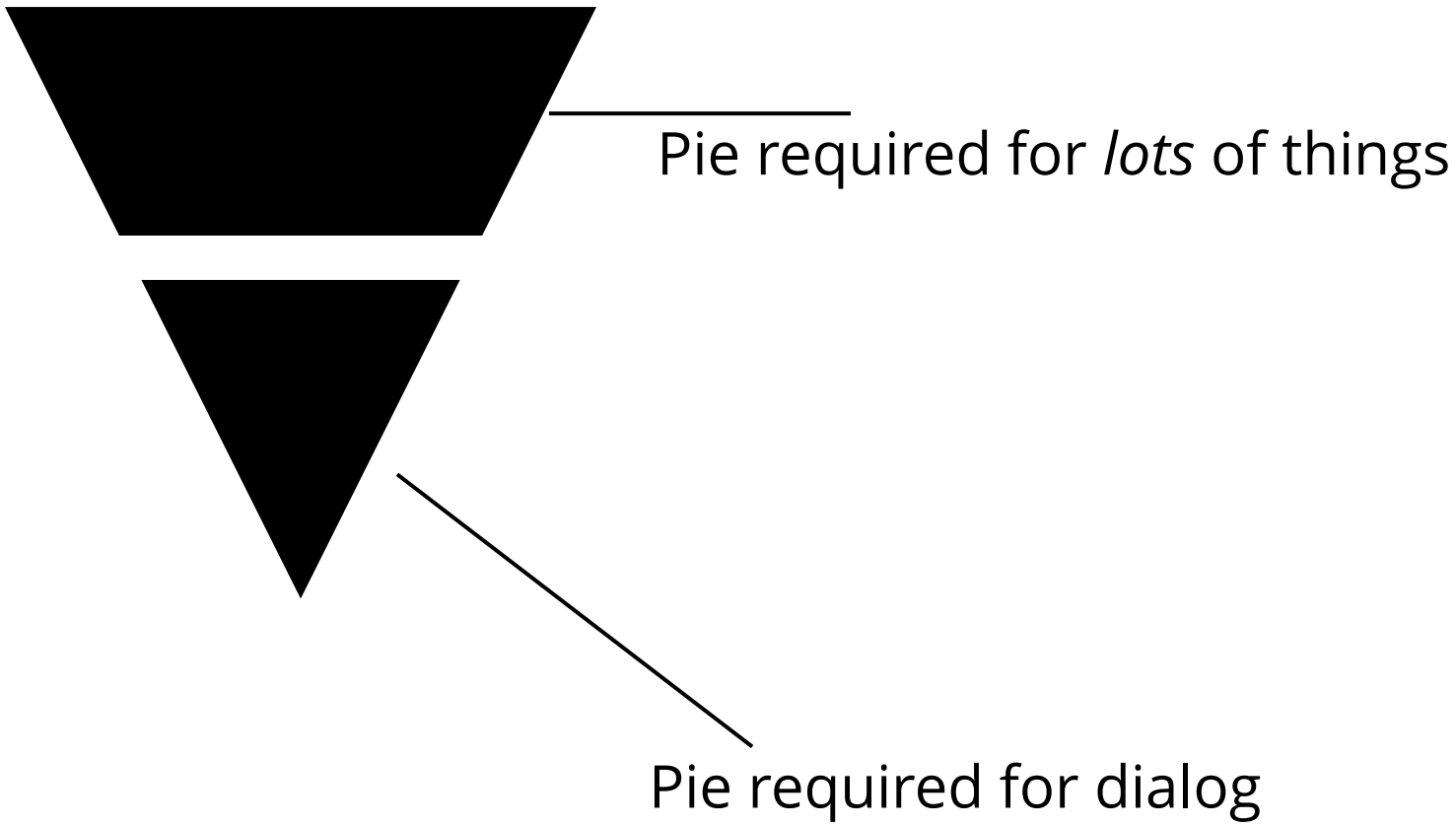
inert:

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

Necessary for dialog

but *also u*seful 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

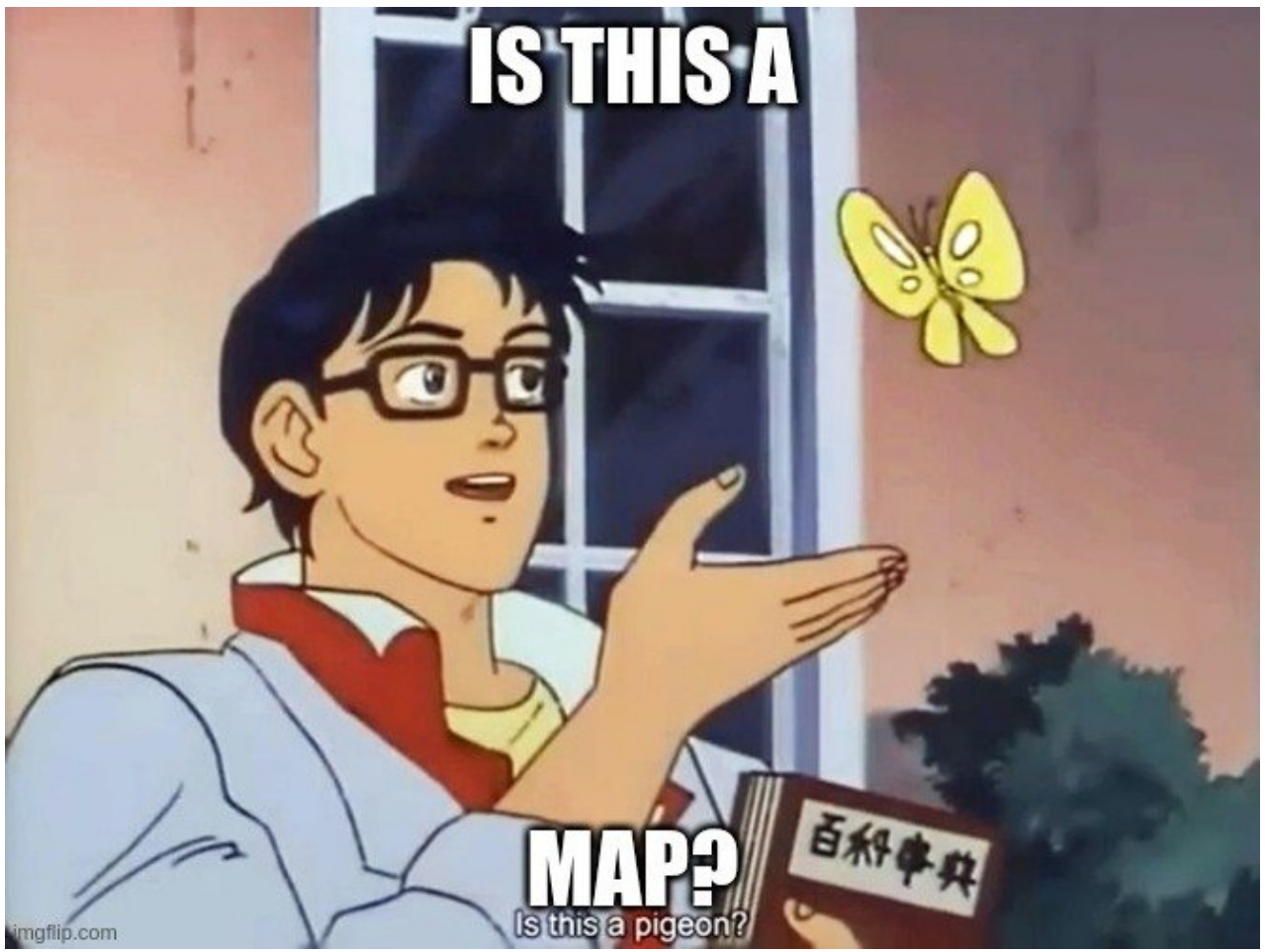
Should the web have maps?

...yes?

We need *so much more pie*

Just bring more pie.

All of these things apply...


IS THIS A
MAP?
Is this a pigeon?
imgflip.com

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.