

For the Users

Mark Nottingham December 2020

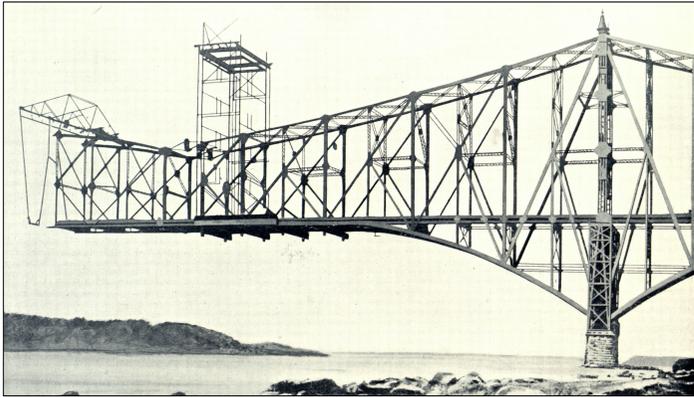
Hi everyone. Today I want to talk about making technology work for its users, and what that means to the technology community. But first I want tell a story about pre-Internet engineering.



Theodore Cooper was an engineer, but he didn't work with Rust, Intel processors or HTTP - he worked with iron, steel and wood. His specialty was building bridges. In fact, he was one of the most well-respected bridge builders at the turn of the century, and also was at one point director of the American Society of Civil Engineers.

He was also a bit of a cowboy, and I think we can recognise some aspects of what we call 'engineering' in him. He graduated as a civil engineer at the age of 19, joined the Navy for a spell during the American Civil War, and then worked for a string of engineering companies, overseeing the construction of a succession of bridges.

At one point he went without sleep for sixty-five hours straight to deal with an emergency, and another time he sent a wire to his boss at midnight because the arch of a bridge was rupturing, and he needed advice on how to avoid it. He didn't have StackOverflow.



One of Mr. Cooper's biggest and best known projects was the First Quebec Bridge in Quebec City, seen here under construction in 1907. It took a long time to design, because the Quebec Bridge Company didn't have flexibility in its budget, and this was a big span. So large that it would be, once completed, the world's longest spanning bridge.

Another problem was that Edward Hoare, the company's chief engineer, had never worked on a cantilevered bridge longer than 300 feet, and this bridge was planned to span 1,600 feet. So they were thrilled to get the famous Theodore Cooper on as a Consulting Engineer, and he quickly took charge of the project.

One of the first things he did was recommend that the bridge be extended to 1,800 feet, so that the piers wouldn't catch as much ice in the winter, speeding up construction as well. The extra steel in the bridge cost more money, and the budget was tight, so Cooper modified the specifications to use less. No one questioned the modified design; after all, they had a 10x bridge engineer on their hands.

Then they started to build.



It's called the First Quebec Bridge because on 29 August 1907, it collapsed while still under construction. 75 workers were killed, and 11 injured. A Royal Commission investigated the disaster and found that Cooper and one of his peers were responsible for the failure, because of their errors in judgement, and lack of supervision over the project. They hadn't bothered to do the math on how much steel the bridge actually needed as its design evolved, instead relying on Cooper's star power to assure success.



The First Quebec Bridge disaster led to a lot of soul searching and discussion about the power that engineers have. Canada and the United States already had professional societies for civil engineers, but ethics hadn't been a prominent part of the discussion until then.

This was in a time when technology was not only putting the workers building it at risk, but was also changing society in fundamental ways. Railroads were originally a technical achievement, but deployed at scale they amassed considerable power and wealth into a few hands, and by the late 1800's necessitated a whole new form of law, Anti-Trust, to protect consumer welfare and combat market domination.

Engineers had started to become thoughtful about their impact on the world, and one of the outcomes was a ring made of iron, like this one, that new engineers received upon taking an oath.

I am an Engineer.

In my profession I take deep pride. To it I owe solemn obligations.

As an engineer, I, (full name), pledge to practice Integrity and Fair Dealing, Tolerance, and Respect, and to uphold devotion to the standards and dignity of my profession, conscious always that my skill carries with it the obligation to serve humanity by making best use of the Earth's precious wealth.

As an engineer, I shall participate in none but honest enterprises. When needed, my skill and knowledge shall be given without reservation for the public good.

In the performance of duty, and in fidelity to my profession, I shall give the utmost.

- Obligation of an Engineer

There are a few versions of that oath; the Canadian one was written by Rudyard Kipling. This is the US version, and bestows a duty upon engineers to be conscious of their obligation to serve humanity.

To paraphrase, engineers create technology, and technology gives leverage to do new things. That leverage is power, and with power comes responsibility.

Or, to put it another way, hubris is always a danger for engineers -- whether you're building a bridge or building a social network. Our actions have affects on the real world.

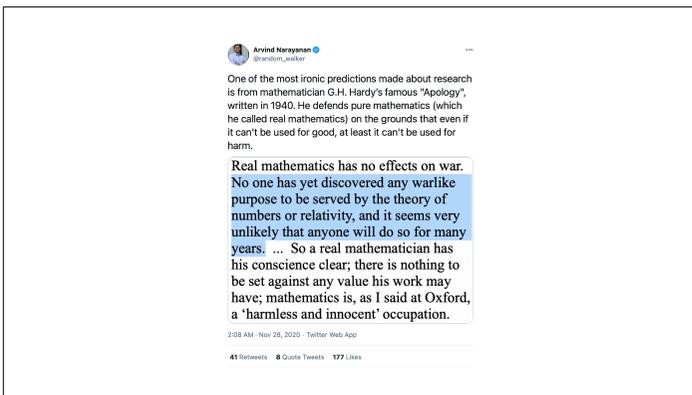


This is a Timex Sinclair 1000 -- my first computer. It cost my parents US\$99.95 and with 2K of memory, a membrane keyboard, and a cassette tape for longer-term storage, it spent a lot of time hooked up to our TV, where I quickly learned the ins and outs of BASIC.

For me as a 11 year old, it offered a window into a whole new world -- one that I could control absolutely.



It's a well-worn trope -- nerds like computers because with them, we don't have to deal with the real world. There is some truth to it, I suspect.



For some, working on tech is a retreat into purity -- a 'harmless and innocent' place precisely because of its disconnection from the 'real world'.

“Tech should not be political.”

- lots of tech people

I think these are reasons why we see assertions like this a lot - that tech should not be political.

It's tempting to think that tech is pure; that we can separate it from the world that it's used in.

“MOAR TECH will fix \$problem.”

- fewer tech people these days

The opposite side of that coin is the notion that tech can and will improve anything -- technical utopianism. Fewer people ascribe to this belief publicly nowadays, because I think most realise how absurd it sounds. But old habits die hard, and I think more than a few tech people have a bias towards more tech as a solution to any given problem.

Technology itself is not political.

The processes of creating technology are often political.

The impact of technology on society is political...

... but hard to predict.

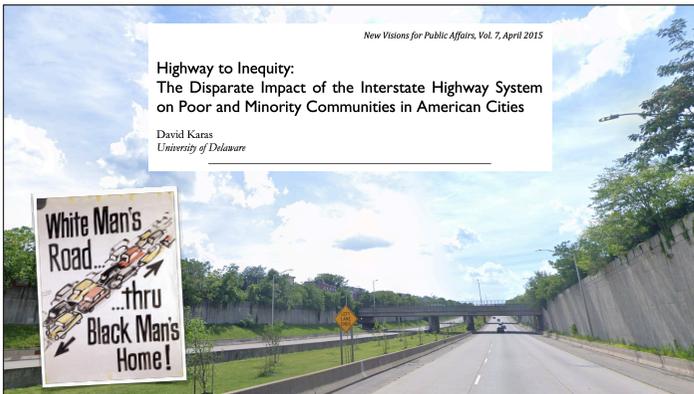
The truth is that tech itself isn't very political. However, the processes we use to create it often are, and the impact it has on society is almost always political in some fashion. It's also hard to predict.

In • fra • struc • ture

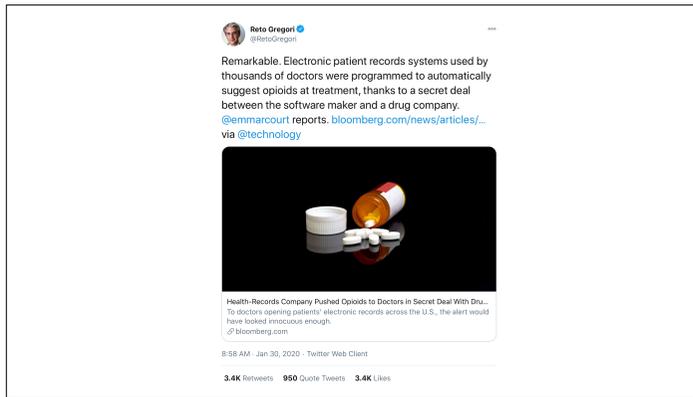
Or, to put it another way, tech people often think of themselves as building infrastructure. Infrastructure itself isn't political; it's just infrastructure. However, the process of creating it as well as its effects on society are often intensely political.



Take roads. Roads aren't political; they're just roads. But it often takes a significant community consultation process spanning years to plan and build a road, because its effects on people can be profound. Building roads changes the way a society organises itself, and you'll notice that if you compare places with different approaches to road-building.



A road can also have more direct and negative effects on a community. The road you see pictured here is I-170 in Baltimore, near where I grew up. It splits a black community in two, as many other roads did when the Interstate Highway System was built in the United States. Infrastructure isn't inherently political, but its effects certainly can be.



Which brings us to Internet technology and infrastructure. Open standards, open source and what your company might be doing.

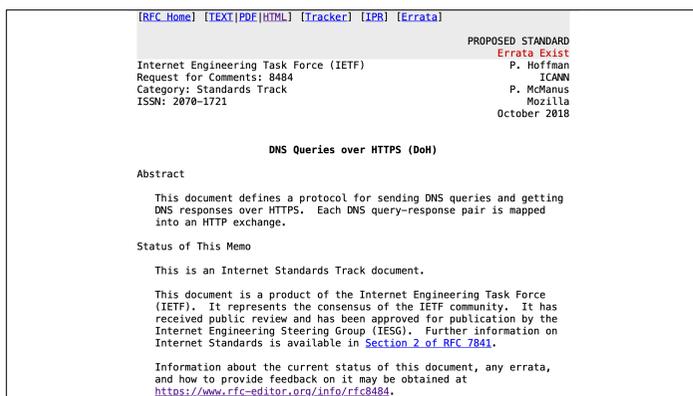
There are plenty of examples of how they can have deep and uneven impacts on society, but let's go through just a few.

Here's a report of a software company who made a secret deal with a drug company to automatically suggest their opioids as treatment. Engineers designed the feature, wrote that code, and tested it (probably).



Here's a more subtle case. An engineer was given a task to design an algorithm to crop images, and someone found that it appears to favour white faces over Black ones. They used Machine Learning, and it's still not clear how biased the algorithm actually is, and more importantly, why.

It's hard to support the notion that the engineering choices here don't have significant effects on society. Should the engineer have anticipated this? What processes and habits would make that consideration more likely?



Similar aspects of Internet Standards are also increasingly hard to ignore. DNS over HTTPS encrypts DNS requests so that the network can't see them. That's great if you're a user in a coffee shop, or maybe if you're a citizen of a repressive regime, but it's freaking out a lot of enterprise network operators, because they use DNS to detect malware (and maybe control what their users do). Do networks have the right to decide what people do with them, and to see what they're doing? Is that part of the design of the Internet?

“Move fast and break things.”

Overall, there's a tendency in our industry to respond to this by burying our heads in the sand and pretend that tech is special; that we can still, in the words of Mark Zuckerberg, "move fast and break things."



But the rest of the world has noticed, and they're not very happy with tech. This shouldn't surprise us; the Internet is no longer a place dominated by technical people; it's where more than half of the world's population goes for information, services, entertainment, work and civic participation.



And people are trying to figure out how to rein in the worst effects of the rapid changes that technology has brought. For example, competition authorities have shown intense interest in big Internet platforms over the last few years.

If you haven't already, skim read one or two of these reports, and ask yourself whether their conclusions about market power and abuse are reasonable. To many, the big question here isn't whether big platforms have market power, or that in many cases they're abusing it; the question is what the right remedies are.

Coronavirus Vaccine Tracker

By Jonathan Corum, Sui-Lee Wee and Carl Zimmer Updated November 23, 2020



Vaccines typically require years of research and testing before reaching the clinic, but scientists are racing to produce a safe and effective coronavirus vaccine by next year. Researchers are testing 54 vaccines in clinical trials on humans, and at least 87 preclinical vaccines are [under active investigation](#) in animals.



Regulation is Coming.

Remember that 'move fast and break things' is not normal in many places. Consider for a moment how much society needs -- not just wants -- a coronavirus vaccine right now. Even though many, many candidates exist right now, they aren't allowed on the market; they still have to go through rigorous testing to prove that they're safe, and they work.

It takes months and years; you don't ship a vaccine in a six-week sprint.

Some people are asking why the same rigour isn't applied to digital platforms, especially when they can be used to sway elections.

And, while much of the attention is on big platforms for now, it's becoming clear that governments, civil society and the public at large are no longer happy to leave the tech world alone in defining the details of how an increasing amount of everyone's time, attention and money is spent. Some people want to control what's available online and how it's exposed.

Regulation of technology is coming. The question is what form it will take, and how the tech community will interact with it.

In the past, there have been a number of adversarial encounters between governments and various tech communities, especially regarding topics like encryption. This all-or-nothing approach pits regulation by law against regulation by architecture -- that is, the code that we write.

Often, when a regulator says that they've consulted with the tech community, it means that they've held meetings with policy specialists from the big platform companies; the broader open source and internet technical communities are left out.

These aren't healthy interactions, and they don't bode well for the future of tech.

Norms
Markets
Architecture
Law

But it doesn't have to be that way. Lawrence Lessig observes that there are four modalities of regulation -- by norms, markets, architecture and law.

Each of these is a form of constraint that could work on its own, but they're most effective -- and less harmful -- when they work together.

For example, smoking kills people, but social constraints on smoking, the price of cigarettes, the limits of how a smoker can use cigarettes due to things like smoke and fire risk, and laws against things like smoking in restaurants and selling cigarettes to minors all work together to reduce the harm of smoking while still balancing the rights of smokers. They are much more effective than a simple ban on cigarettes, which would likely be circumvented.

Norms
Markets
Architecture
Law

Net Etiquette, Communities
Advertising, Network Peering
Standards, Code
GDPR, ePrivacy, CDA, DCMA

And applied to the Internet, we can see net etiquette (or the lack thereof) as a norm, advertising as a funding model as one of the big market constraints (for better or worse), and both standards and code -- especially open source -- as the architecture.

The law has already shown some interest in technology, with things like the GDPR and ePrivacy directive in Europe, and the CDA and DMCA in the United States. And, despite how scary some of the potential interventions being talked about sound, lawmakers in most jurisdictions are acutely aware that anything they do is likely to have negative effects, and so they're very receptive to relying on the other modalities where possible.

A Choice.

So I think that technical people and people who care about the continued health of Internet-related technology have a choice.

We can continue to move fast and break things, and have legal regulation imposed upon technology without much say.

Or, we can attempt some regulation by architecture and through norms in our community, to align technology's capabilities with society's needs. It may not prevent more legal regulation, but it might soften its negative effects.

For the Users.

One of the ways we can align technology with society's needs is to explicitly prioritise the needs of end users over anyone else -- actively working to serve them better with technology, not just getting more time, attention or money from them. This idea has turned out to be central to a lot of efforts recently.

In case of conflict, consider users over authors over implementors over specifiers over theoretical purity. In other words costs or difficulties to the user should be given more weight than costs to authors; which in turn should be given more weight than costs to implementors; which should be given more weight than costs to authors of the spec itself, which should be given more weight than those proposing changes for theoretical reasons alone. Of course, it is preferred to make things better for multiple constituencies at once.

HTML Design Principles, 3.2 Priority of Constituencies

In standards work, an early sign of focus on user needs was the HTML Design Principles' priority of constituencies.

This small piece of text guides many decisions on the Web platform; if you want to change HTML in a way that advantages implementers over users, for example, expect it to be refused.

“Rough consensus and running code.”

Unofficial IETF motto

In the IETF, on the other hand, the historic focus on "rough consensus and running code" conspicuously left users out of the equation; it's who shows up and who writes code that matters in this view (with an emphasis on the code).



"Our job, as technologists, our job as engineers, our job as anybody who cares about the internet in any way, who has any kind of personal or commercial involvement is literally to armor the user, to protect the user and to make that they can get from one end of the path to the other safely without interference."

Edward Snowden, comments at IETF '93

That has started to change recently, especially after Edward Snowden's revelations about pervasive monitoring of network traffic, which angered a lot of IETF participants.

In 2014, Snowden did a remote Q&A with the IETF. One of the things he said to us was that our job, as engineers, is to protect the user. In the IETF, that's a heavy burden, because people depend on our protocols to make the Internet work, and recently we've has been called on to make more decisions about things that could affect users negatively. Things like inserting third parties into network flows.

"rough consensus and running code" doesn't inform those decisions.

Status:	Informational
More info:	Datatracker IPR Info page
Stream:	Internet Architecture Board (IAB)
RFC:	8890
Category:	Informational
Published:	August 2020
ISSN:	2070-1721
Author:	M. Nottingham

RFC 8890 The Internet is for End Users

Abstract

This document explains why the IAB believes that, when there is a conflict between the interests of end users of the Internet and other parties, IETF decisions should favor end users. It also explores how the IETF can more effectively achieve this.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Architecture Board (IAB) and represents information that the IAB has deemed valuable to provide for permanent record. It represents the consensus of the Internet Architecture Board (IAB). Documents approved for publication by the IAB are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc8890>.

We started discussing these topics on the Internet Architecture Board, which is a group of people who are selected to think about long-term issues. The result, after a few years, was RFC8890, "The Internet is for End Users."

This document discusses why the IETF should prioritise user needs, and explores how it might achieve that. It's a call to the engineers in the IETF to protect users.

I've already talked about the why, so let's briefly cover the how, and then see if we can extend it to other places like Open Source and tech companies.

What's in the interest of end users?

Probably the most obvious and hardest question in this area is how to know what's best for end users.

However, it's the wrong question. The IETF (or any other technical body) shouldn't put itself in the place of deciding what's best for everyone; that's tech paternalism, and we don't have (and shouldn't have) the authority to impose our will on the rest of the world.

On the other hand, we shouldn't just blindly rubber-stamp every specification that comes our way; that would quickly lead to the parties with the lowest expectations setting the standard of what happens on the Internet.

Identify Harms

The For the Users RFC argues that instead of trying to determine what's best for users, we should have more concrete and achievable goals: in particular, we need to get better at identifying harm to end users, especially unintentional harm.

That means thinking about the consequences of our design choices more carefully, and understanding how technology is used - and misused - in lots of different situations.

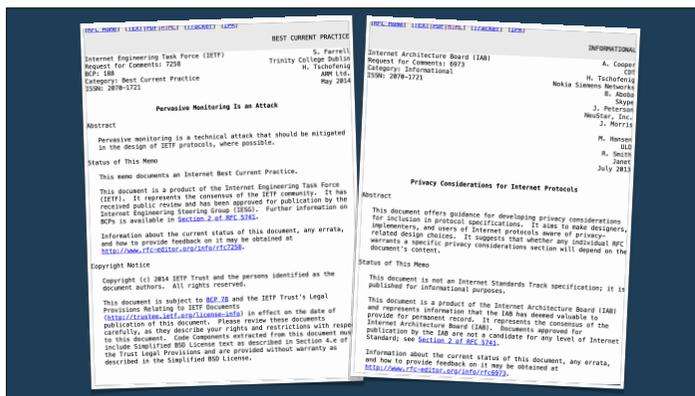
"But the plans were on display..."
"On display? I eventually had to go down to the cellar to find them."
"That's the display department."
"With a flashlight."
"Ah, well, the lights had probably gone."
"So had the stairs."
"But look, you found the notice, didn't you?"
"Yes," said Arthur, "yes I did. It was on display in the bottom of a locked filing cabinet stuck in a disused lavatory with a sign on the door saying 'Beware of the Leopard.'"

- Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

To get that understanding, we need to have more diverse interactions with the communities that use our protocols.

However, participating in the IETF is really difficult for non-technical people. In fact, it's not just difficult, it's unreasonable to expect it.

That's why the For the Users RFC encourages us to leave the comfort of our mailing lists and meetings to gather input and feedback; we shouldn't expect people to come to us. Instead, we need to find where the affected communities are and participate there.



Even with more insights about how our decisions affect end users, there are going to be situations where we'll need to balance many identified harms against each other, or where there are conflicting priorities amongst different groups of users.

There's no magic recipe to make the right decision in these cases, but we can get some help. By applying a set of agreed-to principles, we're more likely to be aligned with the architecture of the Internet and get outcomes that are consistent. That makes the development of those principles critical. Luckily, the IETF already has a number of user-centric principles documented.

For example, we have strong guidance on things like privacy, and fighting surveillance. The focus is on providing an Internet with the properties that people have come to expect. As attacks get more sophisticated, we roll out more sophisticated protections, which is why you

see protocols like DNS over HTTPS being deployed.

However, we're not done; we need to continuously developing user-centric principles.

Create User-Agents

Another area of focus for the RFC is in creating user-centric systems.

For example, the Web is designed so that browsers actively try to represent user needs, thanks to their built-in role as mediators between content and users.

Comparatively, the Internet of Things is a trash fire, where users don't have anyone on their side, and services get to decide how much to respect user privacy and autonomy on a case-by-case basis.

It's not that the Web is perfect, by any means, and browsers make a lot of tradeoffs in their decisions. The point is that systems with incentives for user-positive behaviour are a huge win, and they should be encouraged.

Deprioritise Internal Needs

Finally, the RFC, implores technologists to put their own needs below those of their users, much like HTML's priority of constituencies does. Yes, writing code and tests is a pain, but that's never an excuse for treating your users badly, or creating a system that's tilted against them.

For the Users: What you can do

So, how does all of this apply to what you do in tech? I think there are three areas that it could -- what we do personally, what open source projects do, and what tech companies do.

In particular, there's been a lot of focus recently on how tech companies and open source efforts treat employees and contributors, with efforts around diversity and inclusion.

These are great developments, but we also need to consider tech's broader effects -- on the people we call users, but who are really our siblings, our children, our friends, our parents, our teachers, and fellow citizens.

Tech has formed a reputation in that world of being used by a few people to gain power over, and money from, many. While contributing to Open Source and improving our own communities is good, it isn't nearly enough.

The JSON License

Copyright (c) 2002 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

People have tried to manage the impact of their technical work before, but often struggle to find effective ways to do it. For example, writing "don't be evil" into your license file might make you feel better, but it doesn't really help anything.



To me, it starts by regularly asking ourselves if we're "the baddies." I've worked in tech for twenty five years, and one of the most common things I've seen is a belief that one's company, an open source project, or a standard is "the good guys" -- without understanding how other people are affected by its actions. That's no longer acceptable in an industry with our amount of impact on society.

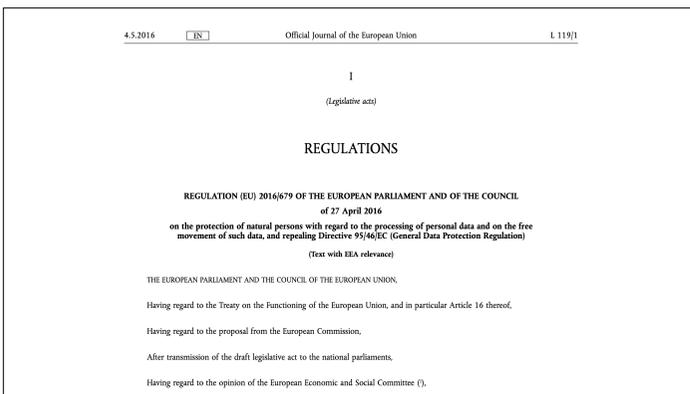


Maybe it means organising a walkout when your employer doesn't live up to your standards.

But more likely, it means considering that your users shouldn't have to completely trust you to use your service. While you might not intend any harm, that doesn't mean that there isn't any.

It means thinking about how what you do might be misused, whose hands it puts power into, and how it affects people. And, in the big picture, what kinds of ecosystems your work is contributing to.

That's something you can bring up at a meeting, or in a design document, or by filing an issue.



Look at the terms of service and other legal agreements for your company or project's site through the eyes of your users and what they want to be assured about.

Tools like the GDPR and data governance efforts are there to protect users -- not annoy you. Work with them, and honour the spirit, not just the letter.

In particular, think about all of the data you keep about users, and whether it's really necessary. Data is immensely powerful, especially in combination. Users' data should remain their property, not yours.

We also have a responsibility to get involved in the larger discussions about the place of tech in society. That's likely to not only involve educating others about tech, but also educating

ourselves about others' perspectives.



And I'd be remiss not to mention the great work that organisations like the EFF and CDT are doing on behalf of users. They need our support.

Does my [project, company, specification] respect user needs and avoid harms if it:

- Performs or enables third party behavioural tracking
- Doesn't use or enable encryption to ensure privacy
- Doesn't give users effective control over how their data is used
- Locks users (or their data) in
- Creates and keeps network effects to itself
- Doesn't give users effective control over the data they consume
- Uses or enables dark patterns to nudge users towards things against their interest
- Has terms of service that are hedged against user interests (e.g., blanket consent)
- Uses or enables fingerprinting or similar techniques for user tracking
- Keeps more data about users (e.g., logs) than necessary (or longer than necessary)
- Uses machine learning to classify people

I know that a lot of this has been pretty abstract, so here are some concrete questions you can ask yourself about your project, your company, or if you happen to work in standards, the specifications you write. It is by no means a complete list, but hopefully you'll get an idea of what I'm talking about here, and how to map it to what you do from day to day.

**We can't separate tech
from the real world.**

I think I've said everything I wanted to about this topic, and maybe a bit more. I hope you've found it interesting and thought provoking.

If I could leave you with just two thoughts, it would be this: we can't separate tech from the real world...

**Be a citizen, not just
an engineer.**

... and we should be fully engaged citizens in society, not just engineers.

Thank you.
