



恭喜

gōngxǐ



<https://scale.qihardware.org>

2019 . Week 9 . Mar 3 - Mar 9

This page left intentionally blank
to power your imagination
of what interesting art, ads,
sponsorship, standard
frontmatter or blank space
should be included in
future editions of scale.

恭喜 gōngxǐ

Gōngxǐ or Gong Hei in Cantonese means "Congratulations", in a celebratory tone.

Gong Hei is used heavily in Cantonese culture during Chinese New Years to congratulate, respectfully wishing one joy. It is often used 恭喜發財 gōngxǐ fācái in Mandarin or often in Hong Kong, Gong Hei Fat Choy, meaning together congratulations for becoming rich.

"GONG HEI" Group Show Launches Oui.Gallery Hong Kong in Central with Maryam Mobasseri, Bao Ho & Rachel Youn

Written on March 5th, 2019 by Rain Wong

恭喜

Hong Kong — Tuesday, March 5, 2019 — Oui.Gallery announced the opening of its new space in Central Hong Kong with its initial show, "GONG HEI". Taking place on International Women's Day, Friday March 8, Oui.Gallery's first show in the new space off Wyndham Street features the artwork of Iranian painter Maryam Mobasseri living in Hong Kong, notable street artist painting abstracted characters on canvas, Bao Ho, and Saint Louis-based Korean-American sculptor Rachel Youn whom uses traditional forms with new materials.

"Oui.Gallery is opening our commercial contemporary art space in Central Hong Kong for easy accessibility for Hong Kong residents and for people traveling through. The goal of the space is to help clients and visitors to envision artwork in their home or office space," said Oui.Gallery Co-founder and Director, Redelle Lee. "Previously, we created art shows for clients and hosted pop-up exhibitions, so our new space, which looks out towards Tai Kwun Centre for Heritage, links into the burgeoning art scene in Central Hong Kong."

"GONG HEI" is a celebration of 1 year of Oui.Gallery in Hong Kong and Saint Louis, two seemingly opposites, both central in their regions.

"Oui.Gallery Hong Kong increases the capacity for cultural bridge building. The new space reaffirms Oui's commitment to emerge local artists in Hong Kong and Saint Louis," said Oui.Gallery Co-founder Jon Phillips. "From the densest city with bar graphs to the sky, to the hollowed out post-industrial smart manufacturing of the emerging midwest, Oui.Gallery presents artifacts from artists from both regions in order to enrich your participation in activating history."

Public Opening: Fri Mar 8, 2019, 6 - 9 PM, Oui.Gallery Hong Kong, 1009 Yu Yuet Lai Building, 43-55 Wyndham St, Central, Hong Kong

Maryam Mobasseri

Maryam Mobasseri (Nooshin Sawyer) was born in 1977, one year before the Islamic revolution of Iran, in Tehran. Her passion for painting was discovered at a very early age. Growing up in Iran during the Iran- Iraq war and being constantly confronted by the ongoing effects of the revolution helped her to find the best way to express her feelings through art. To pursue her lifelong dream to be a painter she quit studying nursing in 1999 and soon after started studying art at Honar University (Art) of Tehran. After completing her master degree in art, she moved to Malaysia and later Hong Kong which exposed her to different cultures. Hong Kong and its diversity has played a very significant role in the development of her artistic style.

Bao Ho

Born and based in Hong Kong, Bao is a self-taught artist known for her freestyle murals and illustrations. She started her professional career in 2015. Bao has a distinctive style: monochromatic art with tightly packed figures including human bodies, animals, flowers and patterns. Her large-scale mural works are transformed into other forms of art through the collaborations with international brands, and she has travelled and worked on projects in all corners of the globe including Australia, Switzerland, UK, France, Taiwan, Mainland China, Japan, Indonesia, Italy and Czech Republic.

Rachel Youn

Rachel Youn is an artist working in St. Louis, MO. They use sculpture and new media to poke fun at hierarchal narratives embedded in objects and lifestyles. Sourcing from home furnishing stores and oriental goods peddled on craigslist, their work collapses notions of authenticity and artifice through the lens of identity. They received their BFA from Washington University in St. Louis in 2017 and has had solo exhibitions at the Bermuda Project (Ferguson, MO), Erica Popp Studios + Gallery and the Millitzer Gallery (St. Louis, MO). Their work has been included in group exhibitions in St. Louis at the Sheldon Art Galleries, Parapet Real Humans, the Luminary and Flood Plain (St. Louis, MO), Open House (Kansas City, MO) and CICA Museum (Gyeonggi-do, South Korea).

Upcoming group exhibitions include Here/There (Portland, OE), Tiger Strikes Asteroid (Los Angeles, CA) and Granite City Art and Design District (Granite City, IL). They are a recipient of the Regional Arts Commission Artist Support Grant and the Vermont Studio Center Fellowship.

About Oui.Gallery

Oui.Gallery is an international contemporary art gallery featuring emerging artists and making innovative shows in Hong Kong and Saint Louis.

Oui.Gallery Hong Kong
1009 Yu Yuet Lai Building
43-55 Wyndham St
Central, Hong Kong

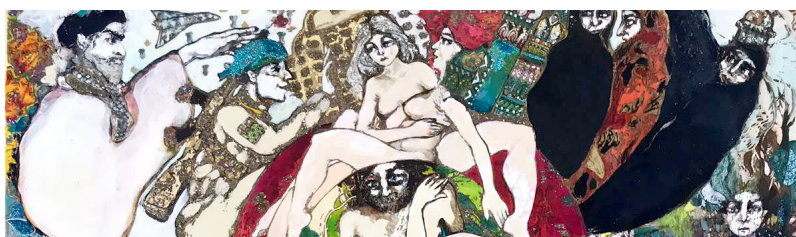
Opening, Friday, March 8, 6-9 PM

Press & Inquiries: info@oui.gallery

Show Flier: [PNG](#) [PDF](#) [SVG](#)

Press Kit: <https://bit.ly/2UeYLhb>

###



You may also enjoy...



Artist Maryam Mobasseri featured at "GONG HEI" Oui.Gallery HK Launch

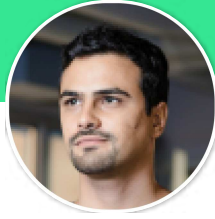


Artist Rachel Youn featured at "GONG HEI" Oui.Gallery HK Launch



Aaron Owens: Artist's Aerial Drone Photos Impact Environment





André Staltz

AN OFF-GRID SOCIAL NETWORK

06 APR 2017

Scuttlebutt is slang for gossip, particularly among sailors. It is also the name of a peer-to-peer system ideal for social graphs, identity and messaging. Scuttlebutt was created by Dominic Tarr, a Node.js developer with more than 600 modules published on npm, who lives on a self-steering sailboat in New Zealand.



Dominic is often offline, but he's still able to use a social network to communicate with his friends such as James Halliday (a.k.a. *substack*), who is also often offline. James has also authored hundreds of npm modules, such as Browserify, and is building a shack with his partner Marina on top of 300-year old lava flows in Hawaii.



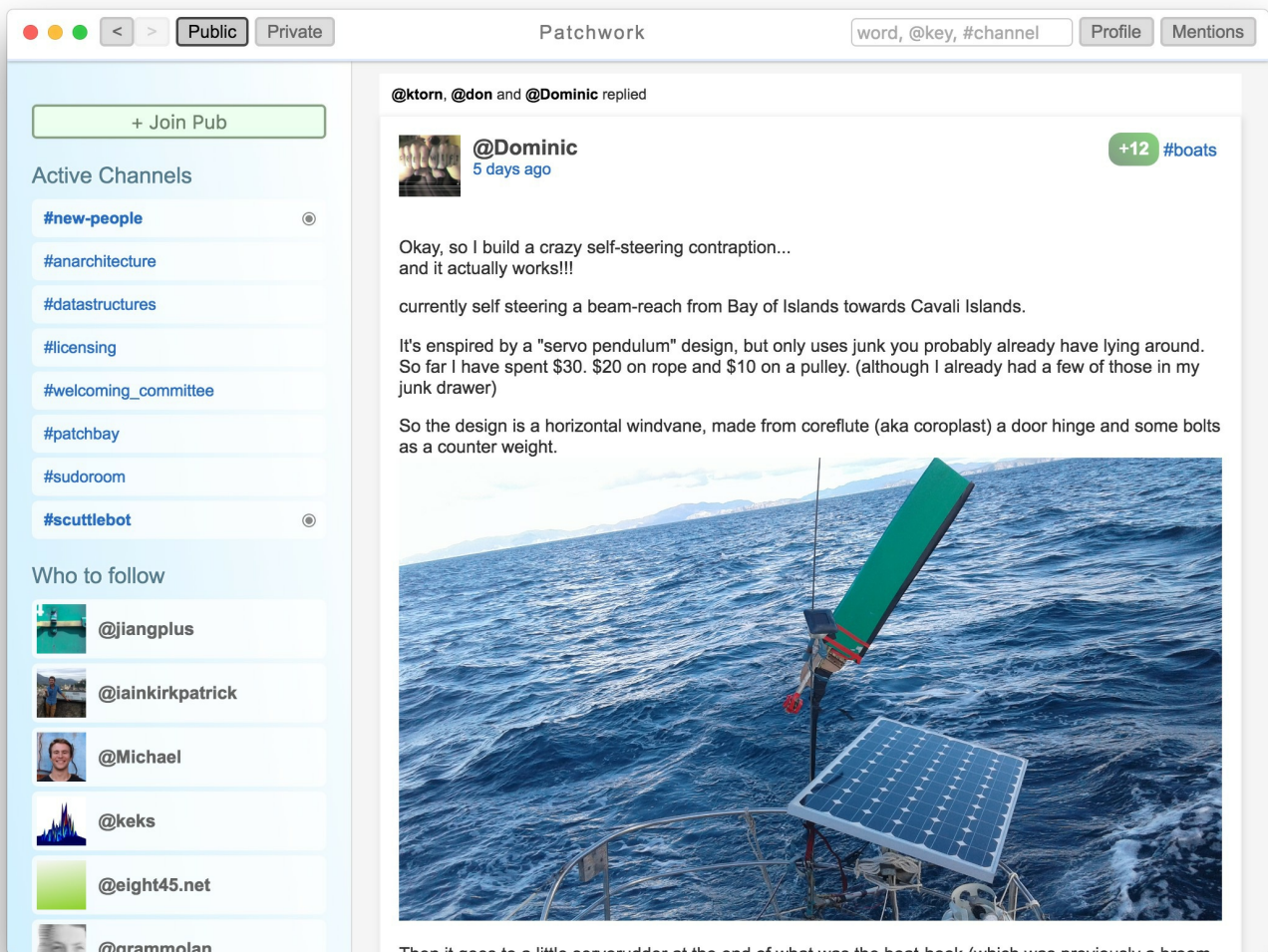
Dominic and James are a few key figures in a community of eccentric open source hackers gathering in a social network independent from mainstream internet. The unique properties of Secure Scuttlebutt (SSB) make it possible for digital information to spread easily even in the absence of Internet Service Providers (ISP) and the internet's backbone. What makes that possible is a *decentralized protocol* based on the mechanics of *word of mouth*.

Scuttlebutt is decentralized in a similar way that [Bitcoin](#) or [BitTorrent](#) are. Unlike centralized systems like PayPal or Dropbox, there is no single website or server to connect when using decentralized services. Which in turn means there is no single company with control over the network.

However, Scuttlebutt differs from Bitcoin and BitTorrent because there are no "singleton components" in the network. When accessing the BitTorrent network, for instance, you need to connect to a Distributed Hash Table (DHT, think of it as a huge round table where anyone can come and take a seat). However, to get access to the DHT in the first place, you need to connect to a bootstrapping server, such as [router.bittorrent.com:6881](#) or [router.utorrent.com:6881](#). These are very lightweight servers which simply introduce you to the DHT. They still depend on

the existence of ISPs and the internet backbone. Also, those systems are concerned about public information. For instance, with Bitcoin, each peer stores the entire log of all transactions ever sent by anyone.

Secure Scuttlebutt is also different to federated social networks like Mastodon, Diaspora, GNU social, OStatus. Those technologies are not peer-to-peer, because each component is either a server or a client, but not both. Federated social networks are slightly better than centralized services like Facebook because they provide some degree of choice where your data should be hosted. However, there is still trust and dependency on third-party servers and ISPs, which makes it possible for administrators of those to abuse their power, through content policies, privacy violations or censorship.



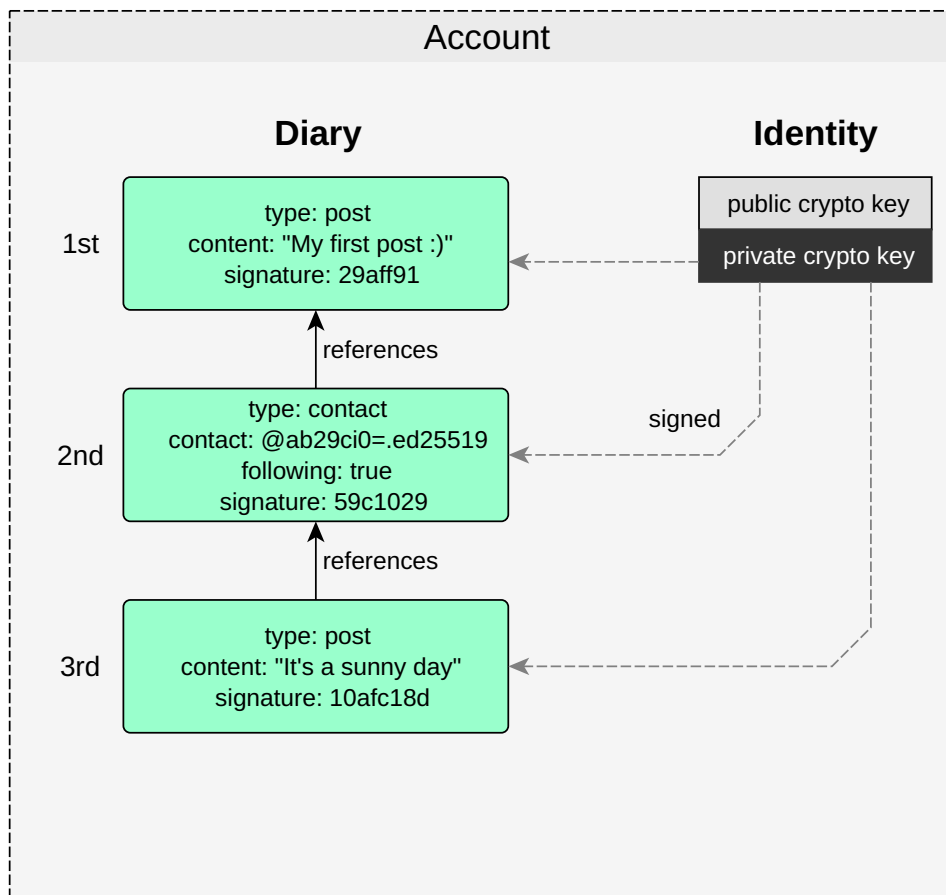
In Scuttlebutt, the “mesh” suffices. With simply two computers, a local router, and electricity, you can exchange messages between the computers with minimal effort and no technical skills. Each account in Scuttlebutt is a diary (or “log”) of what a person has publicly and digitally said.

As those people move around between different WiFi / LAN networks, their log gets copy-pasted to different computers, and so digital information spreads.

What word of mouth is for humans, Scuttlebutt is for social news feeds. It is unstoppable and spreads fast. Once the word is out (just an arbitrary example) that Apple is releasing a new iPhone model, there is no way to restrict that information from spreading. A person may tell that piece of information to any of their friends, and those friends may in turn spread that information onwards.

With typical gossip, however, information deteriorates as it spreads and eventually becomes harmful rumor. Scuttlebutt on the other hand makes word of mouth *secure* with cryptography. Each Scuttlebutt account is comprised of simply two things: an append-only diary and private/public asymmetric crypto keys. An account's identity is its public key. There are no unique usernames, because you can't guarantee two people in separate places from choosing the same username, much like you cannot forbid the name "John Smith" to be given to a newborn in Canada if it is already taken by another person in Australia.

All information a person has published is registered in their diary. Public messages (like in Twitter) are the most common type of message in a diary, but you'll also see "I am friends with that person" type of messages. To 'send' a private message to someone, I simply record a message in my diary, but encrypt it first, so the message isn't plainly readable by anyone who gets their hands on a copy of the diary. Authenticity of diaries is preserved in that all diary entries reference the message that was written before, and then is signed. This prevents tampering and makes replication easier.



Every time two Scuttlebutt friends connect to the same WiFi, their computers will synchronize the latest messages in their diaries. Another way of synchronizing information is to connect to a common Scuttlebutt server, known as "pub", set up by any member in the community. Pubs make information spread faster, and globally, but are totally dispensable. It's even feasible to exchange latest news through sneakernet, using e.g. USB sticks.

This architecture is built so that network connections accurately represent the social graph and word of mouth. Typically with social networks like Facebook or Twitter, the network connections are centralized with their servers. The network architecture looks completely different to social architecture. Most users don't care about this because the network architecture is invisible to them. However, it becomes a real problem once an authoritarian government or even the host company itself takes control over the network architecture in ways that disrupt the social architecture. It is not uncommon for a government to shut down a social network in a country for

days/weeks, affecting how people communicate with each other. This has happened in [Egypt](#), [Cameroon](#), and [Brazil](#).

With Scuttlebutt, the social graph *is* the network architecture, with peer-to-peer infrastructure accurately matching peer-to-peer interactions. It makes communication and the spread of information highly resilient, bringing improvements to freedom of speech with modern information technologies.

This peer-to-peer system has existed for more than two years and brought unique challenges and possibilities. For instance, unique usernames are impossible without a centralized username registry. On the other hand, this questions the need for a login system in the first place: why do you need to “enter” into the service? Scuttlebutt will not have a user registration flow, because such thing makes no sense in that world.

So far, the network has received a dedicated [social network desktop app](#), a [Soundcloud alternative](#), a [Viewer webapp](#), and a [git layer](#) (putting “distributed” back into “distributed version control”). These work seamlessly together: a person using the git layer to push a commit will record that on their diary, which is visible also in the social network app, for their friends. Currently, the community is using this to “eat their own dog food”, coordinating team work and contributing code all on the same platform, without any intermediate company. GitHub being down will rarely be a problem for them.

The platform is being improved constantly, in areas such as: mobile support, an NPM alternative, WebRTC support for browser peers, and even legal transactions in New Zealand. It has proved to work as a platform setting the requirements and examples for a human-centered social network, as Dominic [well described](#):

I wanted an open platform that anyone could build things on. (...) Also, we couldn't realistically plan to just sit down and create an app that everyone wants to use, we need many experiments so that one can succeed, therefore we need a decentralized application platform more than we need any given a decentralized application.

To use Scuttlebutt, I recommend reading the [ssb handbook](#).

If you liked this article, consider sharing ([tweeting](#)) it to your followers.

Become a Patron!

Copyright (C) 2017 Andre 'Staltz' Medeiros, licensed under [Creative Commons BY-NC 4.0](#), translations to other languages allowed. You can make sure that the author wrote this post by copy-pasting [this signature](#) into [this Keybase page](#).

[Back home](#)

[Source code](#)

Hardware, Software, Trustware

presented by Andreas Antonopoulos
on 03/20/2017



The title of this talk is “Hardware, Software, Trustware.” How many of you have been paying attention to this whole business about a fork? Has anybody heard of a fork? There is a culture gap that is being expressed in this fork. And most people look at this and they see the obvious culture gap. Most of the mining happens in China; most of the miners are Chinese; a lot of the development happens in Western nations; a lot of the developers are from Western nations. And at a first glance it looks like the culture gap is some kind of East/West thing.

It’s not. In fact, I think a much better framework for thinking about this is that the culture gap at the center of the debate we’re having today is a culture gap between people who build hardware and people who build software. And those cultures have been diverging since the 1950s.

If you're involved in computer science, if you do hardware or software for a living, you know what I'm talking about. Software goes way back, but not as far back as hardware. Because the very first software was hardware. And if you wanted to use a computer, it ran one program. Some of the first computers ran very *specific* programs, probably one of the earliest examples is the Enigma cracking machine at Bletchley Park, built by Alan Turing to crack the German cryptography during World War II. It did not have software per se. It had *inputs*. You could input the message, and it would try to figure out the key. But it only ran one program. And you could barely really call it a computer. It was an electromechanical device with a bit of electronics; very primitive.

Software started happening in the 60s. And software represented a giant leap forward, because until then if you wanted to reprogram a computer you had to change its wiring. Or you had to flip a lot of switches. Like, *a lot* of switches. Like 10,000 switches in a big bank of switches, and if you got one of them wrong...that was a bit of a problem. Programming in binary, *not* fun.

And out of this, gradually we started having these two cultures emerge, the culture of people who build hardware and the culture of people who build software, and the fundamental difference has to do with the life cycle of development. And that persists to this day.

When you build a hardware device, your life cycle is measured in months if not years. Eighteen to twenty-four months. You design a chip to do something. You make some architecture decisions. These decisions will arrive at the marketplace two years from now. And if you got them wrong, you start again. If you make a mistake in hardware, things happen and you can't issue a patch. If you ship a phone that has a nasty tendency of blowing up, there is no software patch that you can issue that will fix the battery issues. It's out there, in people's pockets, getting hot.

The culture that comes out of that is a very conservative culture. The

planning timelines for building hardware are very long. And they require absolute precision. If you make a mistake you don't just change a line, recompile, try again, issue a patch. You recall \$1 billion worth of silicon *and turn them into scrap*, because you made a mistake. A nanometer-scale mistake.

And at first, software was like that. Because if you programmed, say in Fortran, on a punch card, and you had six minutes of compute time a day, and you spent twenty hours writing your software and punching it into cards, and then you submitted it to the mainframe and during your six minutes the mainframe would go, "Bzzt! Error," now you have to go back, spend a day figuring out why, fixing it, punching it back into cards, and putting it back on the mainframe during your next six-minute window. That's how programming started. And if you were a programmer in those days, the attitude was, "Well, thank God I don't have to flip a thousand switches to do this. This is so much faster! It only takes forty-eight hours to do one life cycle."

And gradually this gap started shrinking and shrinking and shrinking. If you're a really really crap programmer, you don't even plan, design, or think much about your software. You just...put some shit together, hit run, see what happens. Fix it, run, see what happens. Fix it, run, see what happens. You iterate really really fast. The really *great* programmers are not that sloppy. They put *some* thought into every line they change. But all of us have moments when we start hacking. That's where the phrase "hacking" comes from. From Berkeley in the 1970s, where that's how they fixed software. You just hack at it until it works.

But that encourages a mentality where your worst mistake lasts two minutes. And no one gets to see it, because you don't ship that. That's the beginning of a drift apart of two cultures. In hardware, your mistakes are immortalized on silicon. And they might start fires. In software, your mistakes disappear somewhere in a Git log. Nobody really has to look back at the previous commits and see all of the horrible code you wrote. Right, we all do that. If you've done software

programming, that's how most of us program.

That has some very interesting implications for the current debate. Because there's another fundamental and really important difference in the culture between hardware and software. Hardware is all about ship date. You work two years for that day. When it ships. And once it ships, it's out of your hands. You made the right design trade-offs, great. You made the *wrong* design trade-offs, you're gonna be five years behind your competitors until you figure it out again. But that ship date *is* the last date. And once it's out there it's out of your hands.

Look recently at the perennial battle between AMD and Intel. How many of you are familiar with chip architecture at AMD and Intel? About two years ago— Not too many, so I'll make it simple. Two years ago, AMD and Intel started working on their latest-generation chips. And, having a five-year horizon, they had to make some bets. Some trade-offs, some design decisions.

AMD made the bet that most systems that ship have a Graphical Processing Unit. And that for advanced mathematical operations (matrix manipulation, floating point arithmetic, etc.), you would have a development environment that would take that, optimize in OpenGL, and ship it to the graphics card for processing. Because it's 1,000 times faster for that kind of work. So why would you do that on a general-purpose CPU? It didn't seem sensible. So they decided, let's put sixty-four cores on a chip but only give them four or eight floating-point arithmetic units.

Intel said the software is not going to get that good. It's not going to specialize these instructions. Let's put just eight cores on a chip but give *every one* of them a floating-point arithmetic unit.

Two simple trade-offs, right. Two decisions. Is the industry going this way or is it going that way? Are we going to be able to leverage this new technology or not? Will the software catch up with what we're try-

ing to do? Will the architectures of the future look more like this, or like this? Agonizing over that choice, they get a ship date in mind and they ship. Intel got it right. AMD got it wrong. Intel dominated the desktop and server environment, for this cycle. Now, AMD gets to try again, four years later. Three or four years later. That simple trade-off changed the fortunes of that company and the direction of an industry.

We see this in other examples. Let's go for bigger hardware. I'm a fan of aviation; I'm a private pilot. I love that stuff. Boeing, Airbus. Five years or so ago, they made a very important bet.

Airbus said it's mostly going to be about hub-and-spoke connections. You're going to have big hub airports, and they're going to run hundreds or *thousands* of passengers on single routes. So what we need to build is a double-decker aircraft that can seat more people than ever before.

And Boeing said no, it's going to be mostly regional point-to-point. And if we take an aircraft and allow it to extend its range to 12,000 nautical miles, then that choice is the better choice. And they built the Dreamliner.

Boeing was right. Airbus was wrong. Boeing shipped a hundred times more Dreamliners than Airbus, who'll never catch up, quite. So now they get to try that again. But it's going to be a five-year life cycle.

If there's a bug in your software, even in this environment, you can fix it in a matter of hours, ship a new release, done. So one important difference is this idea of ship date. Once you put it out there, you're locked in for a couple of years. And that makes you have a much more conservative attitude.

But, it has a good side. Because you will only encounter one of those ships dates a couple of times a decade. Three times, four times a decade. That's it. With software, however, something else happens.

Your ship date is not the end of your problems. It is the *beginning* of your problems. Because once you ship, maintenance starts. And very quickly it became apparent to the software industry that software behaves a bit like perishable produce. It's got three days of shelf life, after which if you haven't done maintenance it starts rotting on you.

Software gradually degrades. Especially today's software, which is open source software, in a very dynamic environment, with lots of dependencies to third-party libraries. Everything is moving. SSL changes something, we find a new bug, new release. The tolerances of the network change, new release. Berkeley DB doesn't behave the way you expected, new release. And so software is like a never-ending relationship, which you're trapped in. Like, there is no ship date, it's *continuous* shipping. It's continuous maintenance. There is no "and now we're done." There's only "and now our troubles begin!"

This divergence has created a massive culture difference between the culture of miners and the culture of software developers. It is at the root of the current discussion we're having. From the perspective of someone building hardware, of course you're not going to ship a new chip architecture to fix the problem. Just juice up the clock speed. This architecture still has plenty to go, right. It's got room to grow. Change a parameter for God's sake, juice up the clock speed, and we can keep the current architecture and jut ship it.

From a software developer's perspective, if you're looking at the bigger space, it's much better to change the architecture *now* before you have a lot of technical debt, and software crud, and accumulated UTXO, and enormous blockchain sizes on your data store that you have to keep forever. And of course, you're going to be maintaining this shit *either way*...might as well do an architecture change.

This is the fundamental culture difference between the mining community and the software development community in Bitcoin, and not *just* Bitcoin. Bitcoin's just the one that has the strongest, biggest, most

vocal mining community.

But that's not the end. That's just the beginning. Because without even noticing, we now have a completely new category which is going to create a completely new culture. And that is trustware. What is trustware? Trustware is this weird emergent phenomenon that happens when you combine consensus rules that are running and instantiated in software with a backing of hardware deployed on a *global* network, with a diverse set of participants. *All* of the headaches of hardware, *all* of the headaches of software, *and* some new ones.

Now, when you ship is really important. Because if you're making a consensus rule change, you have to coordinate an entire global network. But the ship date is no longer the end of your problems, it's just the *beginning* of the problems because now you have to maintain it forever. And every mistake you make gets baked into the blockchain and has to be carried with you in the consensus rules, forever. In Bitcoin there are no bugs. There are only consensus rules created through tradition.

So how many people here are software developers who've worked in Bitcoin at all? Alright. You probably know about this one. It's a classic. So when you write a multi-signature script and the code gets to the part where it says `opcheck multisig verify`, the code has to go and pop as many keys off the stack as you've defined as the last parameter. `N`, right? And then inspect the signatures which should be `M` (M of N), and do something. Turns out `opcheck multisig verify` pops one extra value. That's a bit of a problem, because if you do a multisig of three keys and it pops four things, there aren't four things on the stack. And if there aren't four things on the stack, you get a stack error, and your script crashes, and your money's no longer spendable.

Now, when that bug happened—because it did happen accidentally back in probably 2011—it wasn't fixed before some people put spendable Bitcoin and redeemed it on the blockchain. And to do that, what

they did was they put what's called a null value—a dummy value. So they go, “Okay. You want to pop four things off the stack, three of which have to be keys and one which is gonna get ignored? Here's *bleh*, key, key, key.” And so you pop “bleh, key, key, key,” throw away the bleh, keep the three keys, it works! Done. But now, that script has to be valid forever. Because every node in Bitcoin validates *everything, forever*. Oops.

Now, developers are writing new versions of the software—multisig is now part of a pay to script hash formula. Guess what. I still pops an extra value. And so I write in my book—there's a big ol' notice and it says, “You will see an extra value in all the redeem scripts. That's because there's a bug. That bug cannot be fixed. It's with us forever.”

Why can it not be fixed? Because the fix is worse than the problem. I mean sure you could fix it. You could just put a thing in the code that says, “from now on just pop three.” And now everybody knows it and they write redeem scripts that just pop three. Fine. No problem. But now you have a piece of code in your blockchain, what is effectively a soft fork, that says, “Before Block X, pop four, *after* Block X, pop three if you see this script.” So that all of the scripts that came before are valid, and all of the scripts that will come next are also valid but don't need that dummy value.

What you've done is you've moved the crud from your script into your code. And now these two, three lines of code need to be maintained forever. What if there's a bug? If you write three lines of code, on average one of them's going to be wrong. So for every line of code you add to the consensus rules, you've got to make sure you don't add a bug. How do you coordinate an international network so that everybody makes sure they change the rules at the same time and you you don't accidentally invalidate transactions? This is the essence of trustware.

The essence of trustware is we are now writing software, that gets

backed by hardware, deployed on a network, and establishes a set of global consensus rules which if you make a mistake on these consensus rules and go out of consensus, you can lose millions. You can get cheated out of transactions. You can be suffering replay attacks or malleation attacks or *all kinds* of other attacks. This is not a game. This is a new software frontier, only it's not software. It's trustware. And trustware is way more complicated than software, or hardware, or software and hardware put together because there's also a global network component that's controlled by independent actors.

Why on earth would we do all this? I mean, it doesn't sound like a fun development exercise. Why are we creating this thing that will have its own culture, trustware developers? Consensus experts. That requires its own deep understanding and analysis and review. Why are we doing this? Because it gives us something amazing. It gives us a centralized platform of trust that is neutral and not controlled by anyone—and that's worth it. But it's *bloody* painful.

And so this is where we are today. Within this network, especially in the case of Bitcoin, we are now seeing a direct conflict. It's not a violent conflict. It's simply a conflict of ideas. It's a disagreement about the future of the network. It's a disagreement about the future of the currency and the future of the consensus rules.

I have to assume good faith. I think both parties see the way forward as the best way forward for Bitcoin. But in technology it's not just a matter of opinion. There *is* truth. Truth means something. There are correct opinions and *incorrect* opinions. There are opinions that match the facts, and ones that don't. It's not a system of belief, it's a system of science.

Unfortunately most of the conversation that's happening really looks like a system of belief. Or more likely a competition of soccer. So, on the one hand you have diehard fans of FC Barcelona and on the other hand diehard fans of Manchester United. There is no right or wrong.

There is no correct answer. There is only my team and your team, and your team is wearing the wrong color. They look silly and they can't play soccer. Clearly. Any intelligent person can see that. Unfortunately that doesn't lead to any scientific conclusions, which is why we're here.

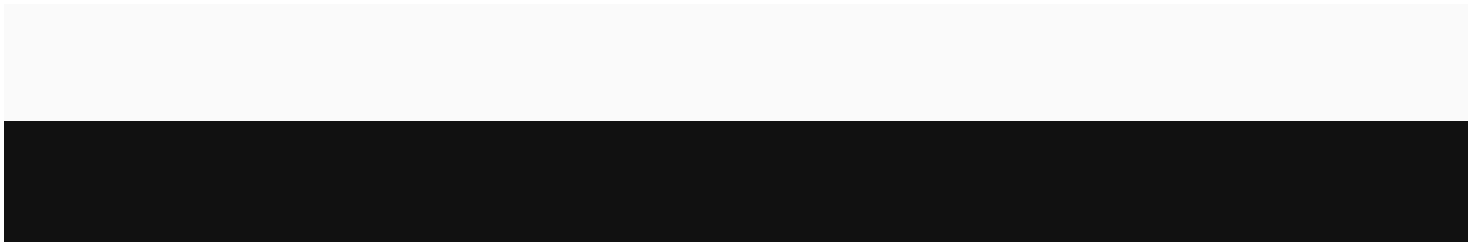
So what we're seeing today is the culmination of a fundamental culture clash between people who primarily build, manage, deploy, and run hardware; and people who fundamentally build, manage, and maintain software. And that's why you'll notice there's some Chinese people on the software side and there's some Westerners on the hardware side, and this is not a culture clash between East and West. It is a culture clash between hardware and software, and from within that a new culture is now emerging. A culture of developers who are building trustware. Who are gradually seeing the nuances and incredible difficulty of building a system of consensus rules that is backed by hardware and deployed on a global network, and that is trustware. Thank you.

Help Support Open Transcripts

If you found this useful or interesting, please consider supporting the project monthly [at Patreon](#) or once [via Square Cash](#), or even just sharing the link. Thanks.

Tags







WIKIPEDIA
The Free Encyclopedia

- [Main page](#)
- [Contents](#)
- [Featured content](#)
- [Current events](#)
- [Random article](#)
- [Donate to Wikipedia](#)
- [Wikipedia store](#)

Interaction

- [Help](#)
- [About Wikipedia](#)
- [Community portal](#)
- [Recent changes](#)
- [Contact page](#)

Tools

- [What links here](#)
- [Related changes](#)
- [Upload file](#)
- [Special pages](#)
- [Permanent link](#)
- [Page information](#)
- [Wikidata item](#)
- [Cite this page](#)

Print/export

- [Create a book](#)
- [Download as PDF](#)
- [Printable version](#)

Languages

- [العربية](#)
- [Čeština](#)
- [Deutsch](#)
- [فارسی](#)
- [Français](#)
-
- [Italiano](#)
-
- [Norsk](#)
- [Português](#)
- [Русский](#)
-

[Edit links](#)

Unix philosophy

From Wikipedia, the free encyclopedia

The **Unix philosophy**, originated by [Ken Thompson](#), is a set of cultural norms and philosophical approaches to [minimalist](#), [modular software development](#). It is based on the experience of leading developers of the [Unix operating system](#). Early Unix developers were important in bringing the concepts of modularity and reusability into software engineering practice, spawning a "software tools" movement. Over time, the



[Ken Thompson](#) and [Dennis Ritchie](#), key proponents of the Unix philosophy

leading developers of Unix (and programs that ran on it) established a set of cultural norms for developing software, norms which became as important and influential as the technology of Unix itself; this has been termed the "Unix philosophy."

The Unix philosophy emphasizes building simple, short, clear, modular, and extensible code that can be easily maintained and repurposed by developers other than its creators. The Unix philosophy favors [composability](#) as opposed to [monolithic design](#).

Contents [\[hide\]](#)

- [Origin](#)
- [The UNIX Programming Environment](#)
- [Program Design in the UNIX Environment](#)
- [Doug McIlroy on Unix programming](#)
- [Do One Thing and Do It Well](#)
- [Eric Raymond's 17 Unix Rules](#)
- [Mike Gancarz: The UNIX Philosophy](#)
- ["Worse is better"](#)
- [Criticism](#)
- [See also](#)
- [Notes](#)
- [References](#)
- [External links](#)

Origin [\[edit\]](#)

The UNIX philosophy is documented by [Doug McIlroy](#)^[1] in the Bell System Technical Journal from 1978:^[2]

1. Make each program do one thing well. To do a new job, build afresh rather than complicate old programs by adding new "features".

2. Expect the output of every program to become the input to another, as yet unknown, program. Don't clutter output with extraneous information. Avoid stringently columnar or binary input formats. Don't insist on interactive input.
3. Design and build software, even operating systems, to be tried early, ideally within weeks. Don't hesitate to throw away the clumsy parts and rebuild them.
4. Use tools in preference to unskilled help to lighten a programming task, even if you have to detour to build the tools and expect to throw some of them out after you've finished using them.

It was later summarized by [Peter H. Salus](#) in *A Quarter-Century of Unix* (1994):^[1]

- Write programs that do one thing and do it well.
- Write programs to work together.
- Write programs to handle text streams, because that is a universal interface.

In their award-winning Unix paper of 1974, Ritchie and Thompson quote the following design considerations:^[3]

- Make it easy to write, test, and run programs.
- Interactive use instead of [batch processing](#).
- [Economy](#) and [elegance](#) of design due to size constraints ("salvation through suffering").
- [Self-supporting](#) system: all Unix software is maintained under Unix.

The whole philosophy of UNIX seems to stay out of [assembler](#).

— [Michael Sean Mahoney](#)^[4]

The UNIX Programming Environment [\[edit\]](#)

In their preface to the 1984 book, *The UNIX Programming Environment*, [Brian Kernighan](#) and [Rob Pike](#), both from [Bell Labs](#), give a brief description of the Unix design and the Unix philosophy:^[5]

Even though the UNIX system introduces a number of innovative programs and techniques, no single program or idea makes it work well. Instead, what makes it effective is the approach to programming, a philosophy of using the computer. Although that philosophy can't be written down in a single sentence, at its heart is the idea that the power of a system comes more from the relationships



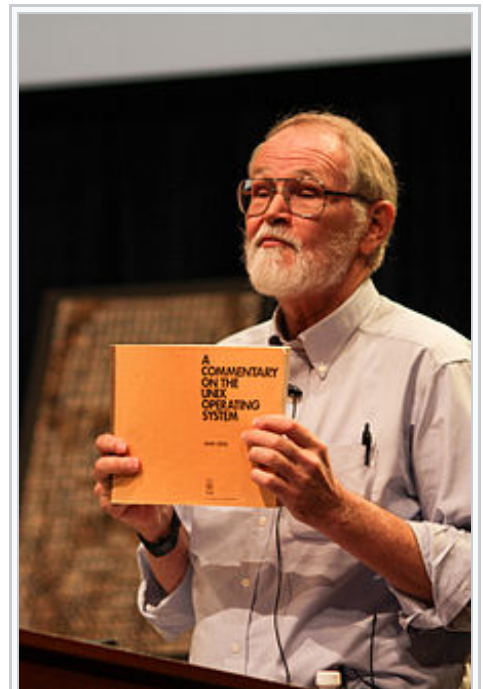
among programs than from the programs themselves. Many UNIX programs do quite trivial things in isolation, but, combined with other programs, become general and useful tools.


The authors further write that their goal for this book is "to communicate the UNIX programming philosophy."^[5]

Program Design in the UNIX Environment [\[edit\]](#)

In October 1984, Brian Kernighan and Rob Pike published a paper called *Program Design in the UNIX Environment*. In this paper, they criticize the accretion of program options and features found in some newer Unix systems such as [4.2BSD](#) and [System V](#), and explain the Unix philosophy of software tools, each performing one general function:^[6]

Much of the power of the UNIX operating system comes from a style of program design that makes programs easy to use and, more important, easy to combine with other programs. This style has been called the use of *software tools*, and depends more on how the programs fit into the programming environment and how they can be used with other programs than on how they are designed internally. [...] This style was based on the use of *tools*: using programs separately or in combination to get a job done, rather than doing it by hand, by monolithic self-sufficient subsystems, or by special-purpose, one-time programs.



[Brian Kernighan](#) has written at length  about the Unix philosophy

The authors contrast Unix tools such as [cat](#), with larger program suites used by

other systems.^[6]

The design of `cat` is typical of most UNIX programs: it implements one simple but general function that can be used in many different applications (including many not envisioned by the original author). Other commands are used for other functions. For example, there are separate commands for file system tasks like renaming files, deleting them, or telling how big they are. Other systems instead lump these into a single "file system" command with an internal structure and command language of its own. (The PIP file copy program found on operating systems like `CP/M` or `RSX-11` is an example.) That approach is not necessarily worse or better, but it is certainly against the UNIX philosophy.

Doug McIlroy on Unix programming [\[edit\]](#)

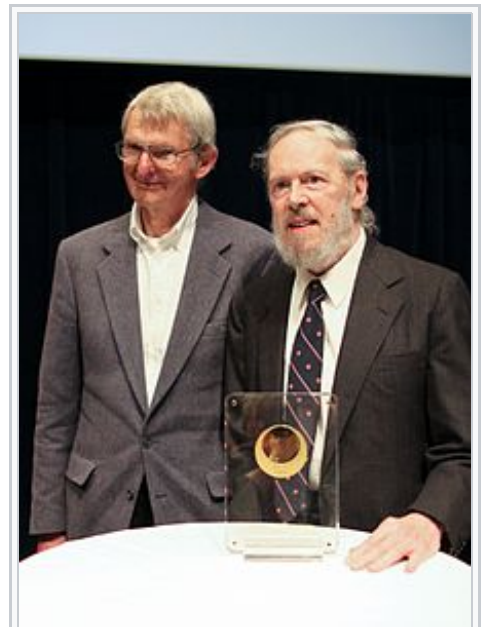
McIlroy, then head of the Bell Labs Computing Sciences Research Center, and inventor of the [Unix pipe](#),^[7] summarized the Unix philosophy as follows:^[4]

This is the Unix philosophy:
Write programs that do one thing and do it well. Write programs to work together.
Write programs to handle [text streams](#), because that is a universal interface.

Beyond these statements, he has also emphasized simplicity and [minimalism](#) in Unix programming:^[1]

The notion of "intricate and beautiful complexities" is almost an oxymoron. Unix programmers vie with each other for "simple and beautiful" honors — a point that's implicit in these rules, but is well worth making overt.

Conversely, McIlroy has criticized modern [Linux](#) as having [software bloat](#), remarking that, "adoring admirers have fed Linux goodies to a disheartening state of [obesity](#)."^[8] He contrasts this with the earlier approach taken at Bell Labs when developing and revising [Research Unix](#).^[9]



Doug McIlroy (left) with Dennis Ritchie

Everything was small... and my heart sinks for Linux when I see the size of it. [...] The [manual page](#), which really used to be a manual page, is now a small volume, with a thousand options... We used to sit around in the Unix Room saying, 'What can we throw out? Why is there this option?' It's often because there is some deficiency in the basic design — you didn't really hit the right design point. Instead of adding an option, think about what was forcing you to add that option.

Do One Thing and Do It Well [\[edit\]](#)

As stated by McIlroy, and generally accepted throughout the Unix community, Unix programs have always been expected to follow the concept of DOTADIW, or "Do One Thing and Do It Well." There are limited sources for the acronym DOTADIW on the Internet, but it is discussed at length during the development and packaging of new operating systems, especially in the Linux community.

[Patrick Volkerding](#), the project lead of [Slackware Linux](#), invoked this design principle in a criticism of the [systemd](#) architecture, stating that, "attempting to control services, sockets, devices, mounts, etc., all within one [daemon](#) flies in the face of the UNIX concept of doing one thing and doing it well."^[10]

Eric Raymond's 17 Unix Rules [\[edit\]](#)

In his book *The Art of Unix Programming* that was first published in 2003,^[11] [Eric S. Raymond](#), an American programmer and open source advocate, summarizes the Unix philosophy as [KISS Principle](#) of "Keep it Simple, Stupid."^[12] He provides a series of design rules:^[1]

- Build [modular](#) programs
- Write readable programs
- Use composition
- [Separate mechanisms from policy](#)
- Write simple programs
- Write small programs
- Write transparent programs
- Write robust programs
- Make data complicated when required, not the program
- Build on potential users' expected knowledge
- Avoid unnecessary output
- Write programs which fail in a way easy to diagnose
- Value developer time over machine time
- Write [abstract programs that generate code](#) instead of writing code by hand
- [Prototype](#) software before polishing it
- Write flexible and open programs
- Make the program and protocols extensible.

Mike Gancarz: The UNIX Philosophy [\[edit\]](#)

In 1994, [Mike Gancarz](#) (a member of the team that designed the [X Window System](#)), drew on his own experience with Unix, as well as discussions with fellow programmers and people in other fields who depended on Unix, to produce *The UNIX Philosophy* which sums it up in nine paramount precepts:

1. *Small is beautiful.*
2. *Make each program do one thing well.*
3. *Build a prototype as soon as possible.*
4. *Choose portability over efficiency.*
5. *Store data in flat [text files](#).*
6. *Use software leverage to your advantage.*
7. *Use [shell scripts](#) to increase leverage and portability.*
8. *Avoid captive user interfaces.*
9. *Make every program a [filter](#).*

"Worse is better" [\[edit\]](#)

Main article: [Worse is better](#)

[Richard P. Gabriel](#) suggests that a key advantage of Unix was that it embodied a design philosophy he termed "worse is better", in which simplicity of both the interface *and* the implementation are more important than any other attributes of the system—including correctness, consistency, and completeness. Gabriel argues that this design style has key evolutionary advantages, though he questions the quality of some results.


For example, in the early days Unix used a [monolithic kernel](#) (which means that user processes carried out kernel system calls all on the user stack). If a signal was delivered to a process while it was blocked on a long-term [I/O](#) in the kernel, then what should be done? Should the signal be delayed, possibly for a long time (maybe indefinitely) while the I/O completed? The signal handler could not be executed when the process was in kernel mode, with sensitive kernel data on the stack. Should the kernel back-out the system call, and store it, for replay and restart later, assuming that the signal handler completes successfully?

In these cases [Ken Thompson](#) and [Dennis Ritchie](#) favored simplicity over perfection. The Unix system would occasionally return early from a system call with an error stating that it had done nothing—the "Interrupted System Call", or an error number 4 (`EINTR`) in today's systems. Of course the call had been aborted in order to call the signal handler. This could only happen for a handful of long-running system calls such as `read()`, `write()`, `open()`, and `select()`. On the plus side, this made the I/O system many times simpler to design and understand. The vast majority of user programs were never affected because they did not handle or experience signals other than `SIGINT` and would die right away if one was raised. For the few other programs—things like shells or text editors that respond to job control key presses—small wrappers could be added to system calls so as to retry the call right away if this `EINTR` error was raised. Thus, the problem was solved in a simple manner.

References [edit]

- *The Unix Programming Environment* by Brian Kernighan and Rob Pike, 1984
- *Program Design in the UNIX Environment* – The paper by Pike and Kernighan that preceded the book.
- *Notes on Programming in C* , Rob Pike, September 21, 1989
- *A Quarter Century of Unix*, Peter H. Salus, Addison-Wesley, May 31, 1994 (ISBN 0-201-54777-5)
- *Philosophy* — from *The Art of Unix Programming* , Eric S. Raymond, Addison-Wesley, September 17, 2003 (ISBN 0-13-142901-9)
- *Final Report of the Multics Kernel Design Project* by M. D. Schroeder, D. D. Clark, J. H. Saltzer, and D. H. Wells, 1977.
- *The UNIX Philosophy*, Mike Gancarz, ISBN 1-55558-123-4

External links [edit]

- *Basics of the Unix Philosophy* – by Catb.org
- *The Unix Philosophy: A Brief Introduction* – by The Linux Information Project (LINFO)
- *The Unix Philosophy*
- *Why the Unix Philosophy still matters* 

Categories: Software development philosophies | Unix

This page was last edited on 5 March 2019, at 15:38 (UTC).

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Cookie statement](#)

[Mobile view](#)





WIKIPEDIA
The Free Encyclopedia

- [Main page](#)
- [Contents](#)
- [Featured content](#)
- [Current events](#)
- [Random article](#)
- [Donate to Wikipedia](#)
- [Wikipedia store](#)

Interaction

- [Help](#)
- [About Wikipedia](#)
- [Community portal](#)
- [Recent changes](#)
- [Contact page](#)

Tools

- [What links here](#)
- [Related changes](#)
- [Upload file](#)
- [Special pages](#)
- [Permanent link](#)
- [Page information](#)
- [Wikidata item](#)
- [Cite this page](#)

Print/export

- [Create a book](#)
- [Download as PDF](#)
- [Printable version](#)

Languages

- [العربية](#)
- [বাংলা](#)
- [Cymraeg](#)
- [Deutsch](#)
- [Español](#)
- [فارسی](#)
- [Français](#)
- [Galego](#)
-
-
- [Hrvatski](#)
- [Bahasa Indonesia](#)
- [עברית](#)
- [Nederlands](#)
- [नेपाली](#)
-
-
- [Português](#)
- [Русский](#)
- [Shqip](#)
- [සිංහල](#)
- [Simple English](#)

Free software movement

From Wikipedia, the free encyclopedia

Not to be confused with [Open-source software movement](#), a related movement.

The **free software movement** (**FSM**) or **free/open-source software movement** (**FOSSM**) or **free/libre open-source software movement** (**FLOSSM**) is a [social movement](#)^[1] with the goal of obtaining and guaranteeing certain freedoms for [software users](#), namely the freedom to run the software, to study and change the software, and to redistribute copies with or without changes. Although drawing on traditions and philosophies among members of the 1970s [hacker culture](#) and academia, [Richard Stallman](#) formally founded the movement in 1983 by launching the [GNU Project](#).^[2] Stallman later established the [Free Software Foundation](#) in 1985 to support the movement.



Richard Stallman circa 2002, founder of the [GNU Project](#) and the free software movement.

Contents [hide]

- [Philosophy](#)
- [Actions](#)
 - [Writing and spreading free software](#)
 - [Building awareness](#)
 - [Ethical equality](#)
- [Legislation](#)
- [Economics](#)
- [Subgroups and schisms](#)
 - [Open source](#)
 - [Stallman and Torvalds](#)
- [Criticism and controversy](#)
 - [Should principles be compromised?](#)
 - [How will programmers get paid?](#)
 - ["Viral" licensing](#)
 - [License proliferation and compatibility](#)
- [See also](#)
- [References](#)
- [Further reading](#)
- [External links](#)

Philosophy [edit]

The philosophy of the movement is that the use of computers should not lead to people being prevented from cooperating with each other. In practice, this means rejecting "[proprietary software](#)", which imposes such restrictions, and promoting [free software](#),^[3] with the ultimate goal of liberating everyone in cyberspace^[4] – that is, every computer user. Stallman notes that this action will promote rather than hinder the progression of technology, since "it means that much wasteful duplication of system programming effort will be avoided. This effort can go instead into advancing the state of the art".^[5]

Members of the free software movement believe that all users of software should have the freedoms listed in [The Free Software Definition](#). Many of them hold that it is immoral to prohibit or prevent people from exercising these freedoms and that these freedoms are required to create a decent society where software users can help each other, and to have control over their computers.^[6]

Some free software users and programmers do not believe that [proprietary software](#) is strictly immoral, citing an increased profitability in the business models available for proprietary software or technical features and convenience as their reasons.^[7]

"While social change may occur as an unintended by-product of technological change, advocates of new technologies often have promoted them as instruments of positive social change." This quote by San Jose State professor Joel West explains much of the philosophy, or the reason that the free source movement is alive. If it is assumed that social change is not only affected, but in some points of view, directed by the advancement of technology, is it ethical to hold these technologies from certain people? If not to make a direct change, this movement is in place to raise awareness about the effects that take place because of the physical things around us. A computer, for instance, allows us so many more freedoms than we have without a computer, but should these technological mediums be implied freedoms, or selective privileges? The debate over the morality of both sides to the free software movement is a difficult topic to compromise respective opposition.^[8]

The Free Software Foundation also believes all software needs [free documentation](#), in particular because conscientious programmers should be able to update manuals to reflect modification that they made to the software, but deems the freedom to modify less important for other types of written works.^[9] Within the free software movement, the [FLOSS Manuals](#) foundation specialises on the goal of providing such documentation. Members of the free software movement advocate that works which serve a practical purpose should also be free.^[10]

Actions [\[edit\]](#)

Writing and spreading free software [\[edit\]](#)

The core work of the free software movement focused on software development. The free software movement also rejects proprietary software, refusing to install software that does not give them the freedoms of free software. According to Stallman, "The only thing in the software field that is worse than an unauthorised copy of a proprietary program, is an authorised copy of the proprietary program because this does the same harm to its whole community of users, and in addition, usually the developer, the perpetrator of this evil, profits from it."^[11]



GNU and Tux mascots around free software supporters at FISL 16 

Building awareness [\[edit\]](#)

Some supporters of the free software movement take up [public speaking](#), or host a stall at software-related conferences to raise awareness of software freedom. This is seen as important since people who receive free software, but who are not aware that it is free software, will later accept a non-free replacement or will add software that is not free software.^[12]

Ethical equality [\[edit\]](#)

Margaret S. Elliot, a researcher in the Institute for Software at the University of California Irvine, not only outlines many benefits that could come from a free software movement, she claims that it is inherently necessary to give every person equal opportunity to utilize the Internet, assuming that the computer is globally accessible. Since the world has become more based in the framework of technology and its advancement, creating a selective internet that allows only some to surf the web freely is nonsensical according to Elliot. If there is a desire to live in a more coexistent world that is benefited by communication and global assistance, then globally free software should be a position to strive for, according to many scholars who promote awareness about the free software movement. The ideas sparked by the GNU associates are an attempt to promote a "cooperative environment" that understands the benefits of having a local community and a global community.^[13]

Legislation [\[edit\]](#)

A lot of lobbying work has been done against [software patents](#) and expansions of copyright law. Other lobbying focusses directly on use of free software by government agencies and government-funded projects.

The [Venezuelan](#) government implemented a free software law in January 2006. Decree No. 3,390 mandated all government agencies to migrate to free software over a two-year period.^[14]

Congressmen [Edgar David Villanueva](#) and Jacques Rodrich Ackerman have been instrumental in introducing free software in [Peru](#), with bill 1609 on "Free Software in Public Administration".^[15] The incident invited the attention of Microsoft Inc, Peru, whose general manager wrote a letter to Villanueva. His response received worldwide attention and is seen as a classic piece of argumentation favouring use of free software in governments.^[16]

In the United States, there have been efforts to pass legislation at the state level encouraging use of free software by state government agencies.^[17]

Economics [\[edit\]](#)

The free software movement has been extensively analyzed using economic methodologies, including perspectives from [heterodox economics](#). Of particular interest to economists is the willingness of programmers in the free software movement to work, often producing higher-quality than commercial programmers, without financial compensation. In his 1998 article "The High-Tech Gift Economy," [Richard Barbrook](#) suggested that the then-nascent free software movement represented a return to the [gift economy](#) building on [hobbyism](#) and the absence of economic [scarcity](#) on the internet.^[18] E. Gabriella Coleman has emphasized the importance of accreditation, respect, and [honour](#) within the free software community as a form of compensation for contributions to projects, over and against financial motivations.^[19]

The Swedish [Marxian](#) economist Johan Söderberg has argued that the free software movement represents a complete alternative to [Capitalism](#) that may be expanded to create a post-work society. He argues that the combination of a manipulation of intellectual property law and [private property](#) to make goods available to the public and a thorough blend between labor and fun make the free software movement a [communist economy](#).^[20]

Subgroups and schisms [\[edit\]](#)

Like many social movements, the free software movement has ongoing internal conflict between the many FOSS organizations ([FSF](#), [OSI](#), [Debian](#), [Mozilla Foundation](#), [Apache Foundation](#) etc.) and their personalities. For instance there is disagreement about the amount of compromises and pragmatism needed versus the need for strict adherence to values.^[21]

Open source [\[edit\]](#)

Main article: [Open-source-software movement](#)

Although commercial free software was not uncommon at the time (see [Cygnus Solutions](#) for example), in 1998 after an announcement that [Netscape](#) would liberate their popular Web browser, a strategy session was held to develop a stronger business case for free software which would focus on technology rather than politics.^[22] After this, [Eric Raymond](#) and [Bruce Perens](#) founded the [Open Source Initiative](#) (OSI) to promote the term "open source software" as an [alternative term for free software](#). OSI wanted to address the perceived shortcomings in ambiguous "free software" term,^{[23][24][25]} and some members of OSI in addition didn't follow the free software movement's focus on non-free software as a social and ethical problem; but instead focused on the advantages of open source as superior model for software development.^[26] The latter became the view of people like Eric Raymond and Linus Torvalds, while Bruce Perens argues that open source was simply meant to popularize free software under a new brand, and even called for a return to the basic ethical principles.^[27]

Some free software advocates use the term [free and open-source software](#) (FOSS) as an inclusive compromise, drawing on both philosophies to bring both free software advocates and open-source software advocates together to work on projects with more cohesion. Some users believe that a compromise term encompassing both aspects is ideal, to promote both the user's freedom with the software and also to promote the perceived superiority of an open-source-development model. This eclectic view is reinforced by the fact that the overwhelming majority of [OSI-approved licenses](#) and self-avowed open-source programs are also compatible with the free software formalisms and vice versa.^[10]

While some people prefer to link the two ideas of "open-source software" and "free software" together, they offer two separate ideas and values. This ambiguity began in 1998 when people started to use the term "open-source software" rather than "free software". People in the community of free software used these separate terms as a way to differentiate what they did. Richard Stallman has called open source "*a non-movement*", because it "*does not campaign for anything*".^[28] Open source addresses software being open as a practical question as opposed to an ethical dilemma. In other words, it focuses more on the development. The open-source movement ultimately determines that non-free software is not the solution of best interest but nonetheless a solution.^{[29][10]}

On the other hand, the free software movement views free software as a moral imperative: that proprietary software should be rejected for selfish and social reasons, and that only free software should be developed and taught to cope with the task of making computing technology beneficial to humanity. It is argued that whatever economical or technical merits free software may have, those are byproducts stemming from the rights that free software developers and users must enjoy. An example of this would be the unlikelihood of free software being designed to [mistreat or spy on users](#).^[30] At the same time, the benefits purveyed by the open-source movement have been challenged both from inside and outside the free software movement. It is unclear whether free and open-source software actually leads to more performant and less vulnerable code, with researchers [Robert Glass](#) and [Benjamin Mako Hill](#) providing statistical insight that this is usually not the case.^{[31][32]}

Regarding the meaning and misunderstandings of the word *free*, those who work within the free software camp have searched for less ambiguous terms and analogies like "[free beer vs free speech](#)" in efforts to convey the intended semantics, so that there is no confusion concerning the profitability of free software. The loan adjective *libre* has gained some traction in the English-speaking free software movement as unequivocally conveying the state of being in freedom that free software refers to. This is not considered schismatic; *libre* is seen as an alternative explanatory device. In fact, free software has always been unambiguously referred to as "*libre software*" (in translation) in languages where the word *libre* or a [cognate](#) is native. In India, where free software has gained a lot of ground,^[33] the unambiguous term *swatantra* and its variants are widely used instead of "free".^{[34][35]}

The free software movement rebuts that while "free" may be prone to confuse novices because of the duplicity of meanings, at least one of the meanings is completely accurate, and that it is hard to get it wrong once the difference has been learned. It is also ironically noted that "open source" isn't exempt of poor semantics either, as a misunderstanding arises whereby people think source code disclosure is enough to meet the open-source criteria, when in fact it is not.^[10]

The switch from the free software movement to the open-source movement has had negative effects on the progression of community, according to Christopher Kelty who dedicates a scholarly chapter to the free software movements in "Theorizing Media and Practice". The open-source movement denies that selectivity and the privatization of software is unethical. Although the open-source movement is working towards the same social benefits as the free software movement, Kelty claims that by disregarding this fundamental belief of the free software advocates, one is destroying the overall argument. If it can be claimed that it is ethical to limit the internet and other technology to only users who have the means to use this software, then there is no argument against the way things are at the moment; there is no need to complain if all morality is in effect.^[36]

Although the movements have separate values and goals, people in both the open-source community and free software community collaborate when it comes to practical projects.^[37] By 2005, Richard Glass considered the differences to be a "serious fracture" but "vitaly important to those on both sides of the fracture" and "of little importance to anyone else studying the movement from a software engineering perspective" since they have had "little effect on the field".^[38]

Stallman and Torvalds [edit]

The two most prominent people associated with the movement, [Richard Stallman](#) and [Linus Torvalds](#), may be seen as representatives of the value based versus apolitical philosophies, as well as the GNU versus Linux [coding styles](#). In the [GNU/Linux naming controversy](#) the FSF argues for the term GNU/Linux because GNU is a longstanding project to develop a free operating system, of which they assert the kernel was the last missing piece.^[39]

Criticism and controversy [edit]

Should principles be compromised? [edit]

[Eric Raymond](#) criticises the speed at which the free software movement is progressing, suggesting that temporary compromises should be made for long-term gains. Raymond argues that this could raise awareness of the software and thus increase the free software movement's influence on relevant standards and legislation.^[40]

Richard Stallman, on the other hand, sees the current level of compromise as a greater cause for worry.^{[21][41][42]}

How will programmers get paid? [edit]

Main article: [Business models for open-source software](#)

Stallman said that this is where people get the misconception of "free": there is no wrong in programmers' requesting payment for a proposed project. Restricting and controlling the user's decisions on use is the actual violation of freedom. Stallman defends that in some cases, monetary incentive is not necessary for motivation since the pleasure in expressing creativity is a reward in itself.^[5] On the other hand, Stallman admits that is not easy to raise money for FOSS projects.^[43]

"Viral" licensing [edit]

The free software movement champions [copyleft](#) licensing schema (often pejoratively called "[viral](#)

licenses"). In its strongest form, copyleft mandates that any works *derived* from copyleft-licensed software must also carry a copyleft license, so the license spreads from work to work like a computer virus might spread from machine to machine. These licensing terms can only be enforced through asserting copyrights.^[44] Critics of copyleft licensing challenge the idea that restricting modifications is in line with the free software movement's emphasis on various "freedoms," especially when alternatives like MIT, BSD, and Apache licenses are more permissive.^{[45][46]} Proponents enjoy the assurance that copylefted work cannot usually be incorporated into non-free software projects.^[47] They emphasize that copyleft licenses may not attach for all uses and that in any case, developers can simply choose not to use copyleft-licensed software.^{[48][49]}

License proliferation and compatibility [edit]

Main article: [Comparison of free and open-source software licenses](#)

FOSS [license proliferation](#) is a serious concern in the FOSS domain due to increased complexity of [license compatibility](#) considerations which limits and complicates source code reuse between FOSS projects.^[50] The OSI and the FSF maintain own lists of dozens of existing and acceptable FOSS licenses.^[51] There is an agreement among most that the creation of new licenses should be minimized at all cost and these created should be made compatible with the major existing FOSS licenses. Therefore, there was a strong controversy around the update of the GPLv2 to the GPLv3 in 2007,^{[52][53]} as the updated license is not compatible with the previous version.^{[54][55][56]} Several projects (mostly of the open source faction^[53] like the [Linux kernel](#)^{[57][58]}) decided to not adopt the GPLv3 while the GNU projects adopted the GPLv3.

See also [edit]

- [GNU Manifesto](#)
- [History of free software](#)
- [Linux adoption](#)
- [Open-source movement](#)
- [Free-culture movement](#)
- [Free Software Foundation](#)
- [Open Source Initiative](#)
- [Software Freedom Conservancy](#)
- [Free Software Movement of India](#)
- [Free Software Foundation of India](#)
- [Free Software Foundation Europe](#)
- [Free Software Movement Karnataka](#)
- [Free Software Foundation Tamil Nadu](#)
- [Swecha](#)
- [Gift economy](#)






















References [edit]

- ↑ [Richard Stallman on the nature of the Free software movement](#) [edit] in 2008 on emacs-devel mailing list.
- ↑ ["Announcement of the GNU project"](#) [edit].
- ↑ ["Use Free Software"](#) [edit]. gnu.org.
- ↑ ["Stallman interviewed by Sean Daly"](#) [edit]. Groklaw. 2006-06-23.
- ↑ **a** **b** ["The GNU Manifesto"](#) [edit]. gnu.org.
- ↑ ["Why free software?"](#) [edit]. gnu.org.
- ↑ ["Copyleft: Pragmatic Idealism"](#) [edit]. gnu.org.
- ↑ ["The Effect of Computerization Movements Upon Organizational Adoption of Open Source"](#) [edit] (PDF). San Jose State University. Archived from [the original](#) [edit] (PDF) on 2006-01-17.
- ↑ ["Free Software and Free Manuals"](#) [edit]. gnu.org.

9. [Free Software and Free Manuals](#) . gnu.org.
10. ^{a b c d} Stallman, Richard. "Why Open Source Misses the Point of Free Software" . GNU Operating System. Free Software Foundation. Retrieved 11 February 2013.
11. ^a "Transcript of Stallman on Free Software" . FSFE. 2006-03-09.
12. ^a "Transcript of Stallman speaking at WSIS" . Ciarán O'Riordan.
13. ^a "Mobilization of software developers" (PDF). Institute for Software Research.
14. ^a "Free software liberates Venezuela" . Free Software Magazine n°10. 2006-02-08.
15. ^a "An English translation of the Free Software bill proposed in Peru" . ^[dead link]
16. ^a "Peruvian Congressman Edgar Villanueva writing to Microsoft about free software" . Archived from the original on 2007-08-29.
17. ^a "Open source's new weapon: The law?" .
18. ^a Barbrook, Richard (1998). "The High-Tech Gift Economy" . *First Monday: Peer-Reviewed Journal on the Internet*. **13** (12). Retrieved July 22, 2018.
19. ^a Coleman (2013), p. 116-7.
20. ^a Söderberg (2007), p. 153-4.
21. ^{a b} Pragmatism in the History of GNU, Linux and Free/Open Source Software Jun 9, 2015 Christopher Tozzi
22. ^a "History of the OSI" . opensource.org.
23. ^a Eric S. Raymond. "Goodbye, "free software"; hello, "open source"" . "The problem with it is twofold. First, ... the term "free" is very ambiguous ... Second, the term makes a lot of corporate types nervous."
24. ^a Kelty, Christopher M. (2008). "The Cultural Significance of free Software - Two Bits" (PDF). Duke University press - durham and london. p. 99. "Prior to 1998, Free Software referred either to the Free Software Foundation (and the watchful, micromanaging eye of Stallman) or to one of thousands of different commercial, avocational, or university-research projects, processes, licenses, and ideologies that had a variety of names: sourceware, freeware, shareware, open software, public domain software, and so on. The term Open Source, by contrast, sought to encompass them all in one movement."
25. ^a Shea, Tom (1983-06-23). "Free software - Free software is a junkyard of software spare parts" . InfoWorld. Retrieved 2016-02-10. "'In contrast to commercial software is a large and growing body of free software that exists in the public domain. Public-domain software is written by microcomputer hobbyists (also known as "hackers") many of whom are professional programmers in their work life. [...] Since everybody has access to source code, many routines have not only been used but dramatically improved by other programmers.'"
26. ^a "Open Source misses the point" . gnu.org.
27. ^a Bruce Perens (17 February 1999). "It's Time to Talk About Free Software Again" . Archived from the original on 16 July 2014. Retrieved 2 April 2015.
28. ^a Gillin, Paul (2016-04-28). "GNU founder Stallman: 'Open source is not free software' - SiliconANGLE" . SiliconANGLE. Retrieved 2017-10-17.
29. ^a Stallman, Richard. "Why 'Open Source' Misses the Point of Free Software | June 2009 | Communications of the ACM" . *cacm.acm.org*. Retrieved 2017-10-17.
30. ^a Stallman, Richard. "Free Software Is Even More Important Now" . *www.gnu.org*. Retrieved 2017-10-17.
31. ^a Glass, Robert L. (2003). *Facts and Fallacies of Software Engineering* . Addison-Wesley. p. 174. ISBN 0-321-11742-5. ISBN 978-0321117427.
32. ^a Benjamin Mako Hill (19 November 2010). "When Free Software Isn't (Practically) Better" . Archived from the original on 3 January 2015. Retrieved 3 April 2015.
33. ^a Bohannon, Mark. "India adopts a comprehensive open source policy" . *opensource.com*. Retrieved 27 August 2015.
34. ^a Truscello, Michael James Anthony (2005). *The Technical Effect: Free and Open Source Software and the Programming of Everyday Life* (Thesis). Waterloo, Ont., Canada, Canada: University of Waterloo.
35. ^a Truscello, Michael (2007). "Free as in Swatantra: Free Software and Nationhood in India" (PDF). Wilfrid Laurier University.
36. ^a *theorizing media and practice* . anthropology of media.
37. ^a "Why "Free Software" is better than "Open Source"" . gnu.org.
38. ^a Richard Glass (2005), "Standing in Front of the Open Source Steamroller", in Joseph Feller;

Brian Fitzgerald; Scott A. Hissam; Karim R. Lakahani, *Perspectives on Free and Open Source Software*, MIT Press, p. 89, ISBN 0262062461

39. ^ ["Linux and GNU - GNU Project - Free Software Foundation \(FSF\)"](#) . Gnu.org. Retrieved 2012-08-13.
40. ^ Eric S. Raymond (2006-07-01). "ESR's "World Domination 201", on the need for more compromise by the free software movement" . catb.org. Retrieved 2016-02-11.
41. ^ ["RMS on the progress of the movement and his worry about compromise"](#) .
42. ^ ["Richard Stallman on "World Domination 201 ""](#) . "I cannot agree to that compromise, and my experience teaches me that it won't be temporary. ... What our community needs most is more spine in rejection of non-free software. It has far too much willingness to compromise. ... To "argue" in favor of adding non-free software in GNU/Linux distros is almost superfluous, since that's what nearly all of them have already done."
43. ^ ["Interview with Richard Stallman"](#) . GNU/LAS s20e10. [Linux action show](#). 2012-03-11. Retrieved 2014-08-22. "RMS: *I'm not gone to claim that I got a way to make it easier to raise money to pay people who write free software. We all know, that to some extent there are ways to do that, but we all know that they are limited, they are not as broad as we would like.*"
44. ^ David McGowan (2005), "Legal Aspects of Free and Open Source Software", in Joseph Feller; Brian Fitzgerald; Scott A. Hissam; Karim R. Lakahani, *Perspectives on Free and Open Source Software*, MIT Press, p. 382, ISBN 0-262-06246-1
45. ^ ["Open Source Licensing Guide"](#) . *New Media Rights*. Retrieved 13 February 2015.
46. ^ Newbart, Dave (2001-06-01). ["Microsoft CEO takes launch break with the Sun-Times"](#) . Chicago Sun-Times. Archived from [the original](#)  on 2001-06-15. (Internet archive link)
47. ^ Kirk St.Amant & Brian Still (2008). "Examining Open Source Software Licenses through the Creative Commons Licensing Model". *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*. Information Science Reference. pp. 382 of 728. ISBN 1-59140-999-3.
48. ^ Byfield, Bruce (2006-08-29). ["IT Manager's Journal: 10 Common Misunderstandings About the GPL"](#) . Retrieved 2008-08-23.
49. ^ Poynder, Richard (21 March 2006). ["The Basement Interviews: Freeing the Code"](#) . Retrieved 5 February 2010.
50. ^ [OSI and License Proliferation](#)  on fossbazar.com by Martin Michlmayr *"Too many different licenses makes it difficult for licensors to choose: it's difficult to choose a good license for a project because there are so many. Some licenses do not play well together: some open source licenses do not inter-operate well with other open source licenses, making it hard to incorporate code from other projects. Too many licenses makes it difficult to understand what you are agreeing to in a multi-license distribution: since a FOSS application typically contains code with different licenses and people use many applications which each contain one or several licenses, it's difficult to see what your obligations are."* (on August 21st, 2008)
51. ^ [license-list](#) 
52. ^ Mark (2008-05-08). ["The Curse of Open Source License Proliferation"](#) . socializedsoftware.com. Archived from [the original](#)  on 2015-12-08. Retrieved 2015-11-30. *"Currently the decision to move from GPL v2 to GPL v3 is being hotly debated by many open source projects. According to Palamida, a provider of IP compliance software, there have been roughly 2489 open source projects that have moved from GPL v2 to later versions."*
53. ^ ^a ^b McDougall, Paul (2007-07-10). ["Linux Creator Calls GPLv3 Authors 'Hypocrites' As Open Source Debate Turns Nasty"](#) . informationweek.com. Archived from [the original](#)  on 2008-04-13. Retrieved 2015-02-12. *"[...]the latest sign of a growing schism in the open source community between business-minded developers like Torvalds and free software purists."*
54. ^ ["Frequently Asked Questions about the GNU Licenses – Is GPLv3 compatible with GPLv2?"](#) . gnu.org. Retrieved 3 June 2014. *"No. Some of the requirements in GPLv3, such as the requirement to provide Installation Information, do not exist in GPLv2. As a result, the licenses are not compatible: if you tried to combine code released under both these licenses, you would violate section 6 of GPLv2. However, if code is released under GPL "version 2 or later," that is compatible with GPLv3 because GPLv3 is one of the options it permits."*
55. ^ Larabel, Michael (24 January 2013). ["FSF Wastes Away Another "High Priority" Project"](#) . [Phoronix](#). Archived from [the original](#)  on 9 November 2016. Retrieved 22 August 2013. *"Both LibreCAD and FreeCAD both want to use LibreDWG and have patches available for supporting*

the DWG file format library, but can't integrate them. The programs have dependencies on the popular GPLv2 license while the Free Software Foundation will only let LibreDWG be licensed for GPLv3 use, not GPLv2."

56. ^ Chisnall, David (2009-08-31). "The Failure of the GPL" [🔗](#). informit.com. Retrieved 2016-01-24.
57. ^ Kerner, Sean Michael (2008-01-08). "Torvalds Still Keen On GPLv2" [🔗](#). internetnews.com. Retrieved 2015-02-12. *"In some ways, Linux was the project that really made the split clear between what the FSF is pushing which is very different from what open source and Linux has always been about, which is more of a technical superiority instead of a -- this religious belief in freedom," Torvalds told Zemlin. So, the GPL Version 3 reflects the FSF's goals and the GPL Version 2 pretty closely matches what I think a license should do and so right now, Version 2 is where the kernel is."*
58. ^ corbet (2006-10-01). "Busy busy busybox" [🔗](#). lwn.net. Retrieved 2015-11-21. *"Since BusyBox can be found in so many embedded systems, it finds itself at the core of the GPLv3 anti-DRM debate. [...]The real outcomes, however, are this: BusyBox will be GPLv2 only starting with the next release. It is generally accepted that stripping out the "or any later version" is legally defensible, and that the merging of other GPLv2-only code will force that issue in any case"*






Further reading [\[edit\]](#)

- Coleman, E. Gabriella (2013). *Coding Freedom: The Ethics and Aesthetics of Hacking*. Princeton: Princeton University Press. ISBN 0691144613.
- David M. Berry, *Copy, Rip, Burn: The Politics of Copyleft and Open Source*, Pluto Press, 2008, ISBN 0-7453-2414-2
- Johan Söderberg, *Hacking Capitalism: The Free and Open Source Software Movement*, Routledge, 2007, ISBN 0-415-95543-2

External links [\[edit\]](#)

- [What is Free Software?](#) - Essay by Karl Fogel.
- [The Free Software Movement and the Future of Freedom](#), a 2006 lecture by Richard Stallman
- [Free Software Movement intro by FSF](#)
- [The GNU Project Philosophy Directory](#), containing many defining documents of the free software movement
- [An interview with Stallman, "Free Software as a social movement"](#)
- Christian Imhorst, *Anarchy and Source Code - What does the Free Software Movement have to do with Anarchism?*, (licence: GFDL), 2005
- [An anti-DRM campaign](#) - by Bill Xu and Richard Stallman

- [Stallman's Free Software Song](#) 

V · T · E		Free and open-source software
General	Alternative terms for free software · Comparison of open-source and closed-source software · Comparison of source code hosting facilities · Free software · Free software project directories · Gratis versus libre · Long-term support · Open-source software · Open-source software development · Outline	
Software packages	Audio · Bioinformatics · Codecs · Collaboration · Configuration management · Device drivers (Graphics · Wireless) · Geophysics · Health · Mathematics · Operating systems · Programming languages · Routing · Statistics · Television · Video games · Web applications (Content management systems · E-commerce) · Word processors · Android apps · iOS apps · Commercial · Trademarked · Formerly proprietary	
Community	Free software movement · History · Open-source software movement · Organizations · Events	
Licenses	AFL · Apache · APSL · Artistic · Beerware · Boost · BSD · CC0 · CDDL · EPL · Free Software Foundation (GNU GPL · GNU LGPL) · ISC · MIT · MPL · Ms-PL/RL · Python · Python Software Foundation License · Sleepycat · Unlicense · WTFPL · zlib	
	Types and standards	Comparison of free and open-source software licenses · Contributor License Agreement · Copyleft · Debian Free Software Guidelines · Definition of Free Cultural Works · Free license · The Free Software Definition · The Open Source Definition · Open-source license · Permissive software licence · Public domain · Viral license
Challenges	Binary blob · Digital rights management · Hardware restrictions · License proliferation · Mozilla software rebranding · Proprietary software · SCO/Linux controversies · Secure boot · Software patents · Software security · Trusted Computing	
Related topics	Forking · <i>GNU Manifesto</i> · Microsoft Open Specification Promise · Open-core model · Open-source hardware · Shared Source Initiative · Source-available software · <i>The Cathedral and the Bazaar</i> · <i>Revolution OS</i>	
 Book ·  Category ·  Commons ·  Portal ·  WikiProject		

V · T · E		Intellectual property activism
Issues	Artificial scarcity · Copyright infringement · Digital rights management · Gripe site · Legal aspects of file sharing · Mashup (digital · music · videos) · Monopolies of knowledge · Music piracy · Orphan works · Patents (biological · software · software patent debate · trolling) · Public domain	
Concepts	All rights reversed · Alternative compensation system · Anti-copyright notice · Business models for open-source software · Copyleft · Commercial use of copyleft works · Commons-based peer production · Free content · Free software license · Libertarian positions · Open content · Open-design movement · Open Music Model · Open patent · Open-source hardware · Open-source software · Prize system (contests) · Share-alike · Video on demand	
Movements	Access to Knowledge movement · Anti-copyright · Cultural environmentalism · Free-culture movement · Free software movement	
Organizations	Pro-copyright	Copyright Alliance
	Pro-copyleft	Creative Commons · Electronic Frontier Foundation · Free Software Foundation · Open Rights Group · Organization for Transformative Works · The Pirate Bay · Piratbyrå · Pirate Party · Sci-Hub · Students for Free Culture
People	Alexandra Elbakyan · Rick Falkvinge · Lawrence Lessig · Richard Stallman · Peter Sunde · Peter Suber · Aaron Swartz	
Documentaries	<i>Steal This Film</i> (2006, 2007) · <i>Good Copy Bad Copy</i> (2007) · <i>RiP!: A Remix Manifesto</i> (2008) · <i>TPB AFK: The Pirate Bay Away From Keyboard</i> (2013) · <i>The Internet's Own Boy</i> (2014)	

V · T · E		Free-culture and open-source movements
		Commons-based peer production · Crowdsourcing · <i>Gratis versus libre</i> · Open collaboration · Openness ·

Concepts and practices	Key concepts	Participatory culture · Sharism	
		Social peer-to-peer processes Peer-to-peer carsharing · Peer-to-peer lending · Peer-to-peer property rental · Peer-to-peer renting · Peer-to-peer ridesharing	
	Research and science	Citizen science · Open science (Access · Notebook science · Research · Science data · Plan S)	
	Data, information, and knowledge	Free content · Knowledge commons · Open communication · Open knowledge (Content · Data)	
	Communication and learning	Media	Collaborative writing · Democratic media · Open publishing · Participatory media · Peer review
		Education	Open education (Educational resources · University (Admissions · Open-door academic policy))
		Journalism	Citizen media (Citizen journalism and Wiki journalism) · Open-source journalism
	Economy, production, and development	Products	Free and open-source software - FOSS (Free/Libre software · Open-source software) · Open-design movement (Robotics) · Open gaming · Open-source architecture · Open-source hardware
Economic principles		Gift economy · Open innovation · Open patent · Open standard · Sharing economy	
Politics and governance	Open government · Open-source governance		
Organizations	Creative Commons · Free Software Foundation · Open Architecture Network · Openmod initiative · Open Knowledge International · Open Rights Group · Open Source Initiative · Open Web Foundation · P2P Foundation · Pirate Party · PLOS · SPARC		
Activists	Tim Berners-Lee · Alexandra Elbakyan · Cody Wilson · Lawrence Lessig · Peter Murray-Rust · Douglas Rushkoff · Richard Stallman · Peter Suber · Peter Sunde · Aaron Swartz · John Wilbanks		
Projects and movements	DIYbio · Free-culture movement · Free software movement · Open science movement · Open Source Ecology · Open-source software movement · OpenCores · OpenWetWare · Sci-Hub		
	Tools	Licenses (Creative Commons · GPL) · Definition Docs (Free Cultural Works · Free Software · Open · Open Source) · Open Data Indices · Open educational resources · Open music model · Open Web movement	

Categories: [Copyleft media](#) | [Free software culture and documents](#)
[Free software movement](#)

This page was last edited on 7 March 2019, at 05:09 (UTC).

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Cookie statement](#) [Mobile view](#)





WIKIPEDIA
The Free Encyclopedia

- [Main page](#)
- [Contents](#)
- [Featured content](#)
- [Current events](#)
- [Random article](#)
- [Donate to Wikipedia](#)
- [Wikipedia store](#)

Interaction

- [Help](#)
- [About Wikipedia](#)
- [Community portal](#)
- [Recent changes](#)
- [Contact page](#)

Tools

- [What links here](#)
- [Related changes](#)
- [Upload file](#)
- [Special pages](#)
- [Permanent link](#)
- [Page information](#)
- [Wikidata item](#)
- [Cite this page](#)

Print/export

- [Create a book](#)
- [Download as PDF](#)
- [Printable version](#)

In other projects

- [Wikimedia Commons](#)
- [Wikibooks](#)

Languages

- [Afrikaans](#)
- [العربية](#)
- [Asturianu](#)
- [Azərbaycanca](#)
- [বাংলা](#)
- [Български](#)
- [Català](#)
- [Dansk](#)
- [Deutsch](#)
- [Eesti](#)
- [Ελληνικά](#)
- [Español](#)
- [Esperanto](#)
- [Euskara](#)
- [Français](#)
- [ייִדיש](#)

Open-source model

From Wikipedia, the free encyclopedia



This article **possibly contains [synthesis of material](#) which does not [verifiably mention or relate](#) to the main topic.** Relevant discussion may be found on the [talk page](#). *(March 2016)*
([Learn how and when to remove this template message](#))

The **open-source model** is a decentralized [software development](#) model that encourages [open collaboration](#).^{[1][2]} A main principle of [open-source software development](#) is [peer production](#), with products such as source code, [blueprints](#), and documentation freely available to the public. The open-source movement in software began as a response to the limitations of proprietary code. The model is used for projects such as in [open-source appropriate technology](#),^[3] and open-source drug discovery.^{[4][5]}

Open source promotes universal access via an [open-source](#) or [free license](#) to a product's design or blueprint, and universal redistribution of that design or blueprint.^{[6][7]} Before the phrase *open source* became widely adopted, developers and producers used a variety of other terms. *Open source* gained hold with the rise of the [Internet](#).^[8] The [open-source software movement](#) arose to clarify [copyright](#), [licensing](#), [domain](#), and consumer issues.

Generally, open source refers to a [computer program](#) in which the [source code](#) is available to the general public for use or modification from its original design. Open-source code is meant to be a collaborative effort, where programmers improve upon the source code and share the changes within the community. Code is released under the terms of a [software license](#). Depending on the license terms, others may then download, modify, and publish their version (fork) back to the community.

Many large formal institutions have sprung up to support the development of the open-source movement, including the [Apache Software Foundation](#), which supports community projects such as the open-source framework [Apache Hadoop](#) and the open-source [HTTP](#) server [Apache HTTP](#).

Contents [\[hide\]](#)

- [History](#)
 - [1.1 Open source as a term](#)
- [Economics](#)
 - [2.1 Licensing alternatives](#)
- [Open-source applications](#)
 - [3.1 Computer software](#)
 - [3.2 Electronics](#)
 - [3.3 Food and beverages](#)
 - [3.4 Digital content](#)
 - [3.5 Medicine](#)
 - [3.6 Science and engineering](#)
 - [3.6.1 Robotics](#)
 - [3.6.2 Transport](#)
 - [3.7 Fashion](#)
 - [3.8 Other](#)
- [Society and culture](#)
 - [4.1 Government](#)
 - [4.2 Ethics](#)

- हिन्दी
- Hrvatski
- Bahasa Indonesia
- Italiano
- עברית
-
- Latviešu
- Lëtzebuergesch
- Lietuvių
- Bahasa Melayu
- Nederlands
- नेपाली
-
- Norsk
- Norsk nynorsk
- Plattdüütsch
- Português
- Română
- Scots
- Shqip
- Sicilianu
- Simple English
- Slovenčina
- Slovenščina
- کوردی
- Српски / srpski
- Srpskohrvatski / српскохрватски
- Suomi
- Svenska
- தமிழ்
- ไทย
- Тоҷикӣ
- Türkçe
- Українська
- اردو
- יידיש
-

 [Edit links](#)

- 4.3 [Religion](#)
- 4.4 [Media](#)
- 4.5 [Education](#)
- 4.6 [Innovation communities](#)
- 4.7 [Arts and recreation](#)
- 4.8 [Other related movements](#)
- 4.9 [Ideologically-related movements](#)
- 5 [See also](#)
 - 5.1 [Lists](#)
 - 5.2 [Terms based on open source](#)
 - 5.3 [Other](#)
- 6 [References](#)
- 7 [Further reading](#)
 - 7.1 [Literature on legal and economic aspects](#)
- 8 [External links](#)

History [\[edit\]](#)

Main article: [History of free and open-source software](#)

The sharing of technical information predates the [Internet](#) and the personal computer considerably. For instance, in the early years of automobile development a group of capital [monopolists](#) owned the rights to a [2-cycle](#) gasoline-engine patent originally filed by [George B. Selden](#).^[9] By controlling this patent, they were able to monopolize the industry and force car manufacturers to adhere to their demands, or risk a lawsuit.

In 1911, independent automaker [Henry Ford](#) won a challenge to the Selden patent. The result was that the Selden patent became virtually worthless and a new association (which would eventually become the [Motor Vehicle Manufacturers Association](#)) was formed.^[9] The new association instituted a cross-licensing agreement among all US automotive manufacturers: although each company would develop technology and file patents, these patents were shared openly and without the exchange of money among all the manufacturers.^[9] By the time the US entered [World War II](#), 92 Ford patents and 515 patents from other companies were being shared among these manufacturers, without any exchange of money (or lawsuits).^[9]

Early instances of the free sharing of source code include [IBM](#)'s source releases of its [operating systems](#) and other programs in the 1950s and 1960s, and the [SHARE](#) user group that formed to facilitate the exchange of software.^{[10][11]} Beginning in the 1960s, [ARPANET](#) researchers used an open "[Request for Comments](#)" (RFC) process to encourage feedback in early telecommunication network protocols. This led to the birth of the early Internet in 1969.

The sharing of source code on the Internet began when the Internet was relatively primitive, with software distributed via [UUCP](#), [Usenet](#), [IRC](#), and [Gopher](#). [BSD](#), for example, was first widely distributed by posts to comp.os.linux on the Usenet, which is also where its development was discussed. [Linux](#) followed in this model.

Open source as a term [\[edit\]](#)

The term "open source" was first proposed by a group of people in the [free software movement](#) who were critical of the political agenda and moral philosophy implied in the term "free software" and sought to reframe the discourse to reflect a more commercially minded position.^[12] In addition, the ambiguity of the term "free software" was seen as discouraging business adoption.^{[13][14]} The group included [Christine Peterson](#), [Todd Anderson](#), [Larry Augustin](#), [Jon Hall](#), [Sam Ockman](#), [Michael Tiemann](#) and [Eric S. Raymond](#). Peterson suggested "open source" at a meeting^[15] held at [Palo Alto, California](#), in reaction to [Netscape](#)'s announcement in January 1998 of a source code release for [Navigator](#). [Linus Torvalds](#) gave his support the following day, and

Phil Hughes backed the term in *Linux Journal*. [Richard Stallman](#), the founder of the free software movement, initially seemed to adopt the term, but later changed his mind.^{[15][16]} Netscape released its source code under the [Netscape Public License](#) and later under the [Mozilla Public License](#).^[17]

Raymond was especially active in the effort to popularize the new term. He made the first public call to the free software community to adopt it in February 1998.^[18] Shortly after, he founded The [Open Source Initiative](#) in collaboration with [Bruce Perens](#).^[15]

The term gained further visibility through an event organized in April 1998 by technology publisher [Tim O'Reilly](#). Originally titled the "Freeware Summit" and later known as the "Open Source Summit",^[19] the event was attended by the leaders of many of the most important free and open-source projects, including Linus Torvalds, [Larry Wall](#), [Brian Behlendorf](#), [Eric Allman](#), [Guido van Rossum](#), [Michael Tiemann](#), [Paul Vixie](#), [Jamie Zawinski](#), and Eric Raymond. At that meeting, alternatives to the term "free software" were discussed. Tiemann argued for "sourceware" as a new term, while Raymond argued for "open source". The assembled developers took a vote, and the winner was announced at a press conference the same evening.^[19]

"Open source" has never managed to entirely supersede the older term "free software", giving rise to the combined term [free and open-source software](#) (FOSS).

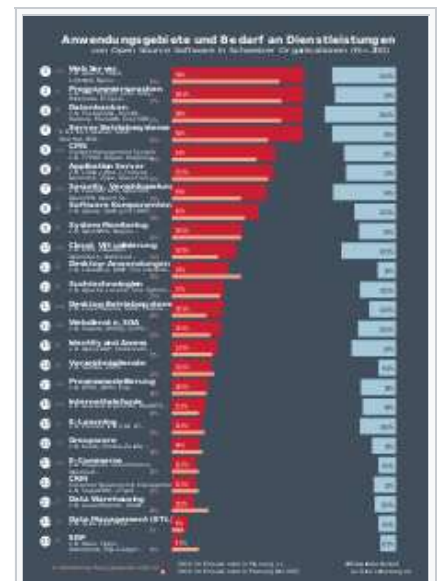
Economics [\[edit\]](#)

Main article: [Open-source economics](#)

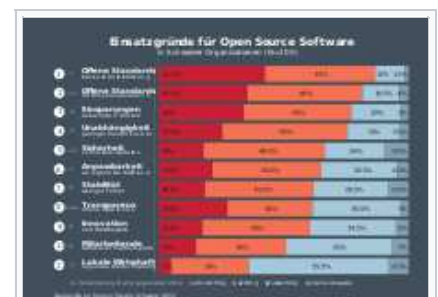
Some economists agree that open-source is an [information good](#)^[21] or "knowledge good" with original work involving a significant amount of time, money, and effort. The cost of reproducing the work is low enough that additional users may be added at zero or near zero cost – this is referred to as the [marginal cost](#) of a product. [Copyright](#) creates a monopoly so the price charged to consumers can be significantly higher than the marginal cost of production. This allows the author to recoup the cost of making the original work. Copyright thus creates access costs for consumers who value the work more than the marginal cost but less than the initial production cost. Access costs also pose problems for authors who wish to create a [derivative work](#)—such as a copy of a software program modified to fix a bug or add a feature, or a [remix](#) of a song—but are unable or unwilling to pay the copyright holder for the right to do so.

Being organized as effectively a "[consumers' cooperative](#)", open source eliminates some of the access costs of consumers and creators of derivative works by reducing the restrictions of copyright. Basic economic theory predicts that lower costs would lead to higher consumption and also more frequent creation of derivative works. Organizations such as [Creative Commons](#) host websites where individuals can file for alternative "licenses", or levels of restriction, for their works.^[22] These self-made protections free the general society of the costs of policing copyright infringement.

Others argue that since consumers do not pay for their copies, creators are unable to recoup the initial cost of production and thus have little economic



Area of application of open source software.^[20]



Survey on the reasons for using Open Source in 200 Swiss organizations.^[20]

incentive to create in the first place. By this argument, consumers would lose out because some of the goods they would otherwise purchase would not be available. In practice, content producers can choose whether to adopt a proprietary license and charge for copies, or an open license. Some goods which require large amounts of professional research and development, such as the [pharmaceutical industry](#) (which depends largely on [patents](#), not copyright for intellectual property protection) are almost exclusively proprietary, although increasingly sophisticated technologies are being developed on open-source principles.^[23]

There is evidence that open-source development creates enormous value.^[24] For example, in the context of [open-source hardware](#) design, digital designs are shared for free and anyone with access to digital manufacturing technologies (e.g. [RepRap](#) 3D printers) can replicate the product for the cost of materials.^[25] The original sharer may receive feedback and potentially improvements on the original design from the [peer production](#) community.

Many open source projects have a high economic value. According to the Battery Open Source Software Index (BOSS), the ten economically most important open source projects are:^{[26][27]}

Ranking	Project	Leading company	Market Value
1	Linux	Red Hat	\$16 billion
2	Git	GitHub	\$2 billion
3	MySQL	Oracle	\$1.87 billion
4	Node.js	NodeSource	?
5	Docker	Docker	\$1 billion
6	Hadoop	Cloudera	\$3 billion
7	Elasticsearch	Elastic	\$700 million
8	Spark	Databricks	\$513 million
9	MongoDB	MongoDB	\$1.57 billion
10	Selenium	Sauce Labs	\$470 million

The rank given is based on the activity regarding projects in online discussions, on GitHub, on search activity in search engines and on the influence on the labour market.

Licensing alternatives [\[edit\]](#)

Further information: [Business models for open-source software](#)

Alternative arrangements have also been shown to result in good creation outside of the proprietary license model. Examples include:^[citation needed]

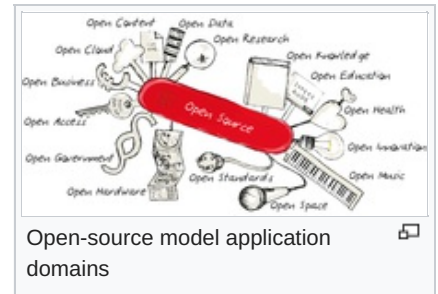
- Creation for its own sake – For example, [Wikipedia](#) editors add content for recreation. Artists have a drive to create. Both communities benefit from free starting material.
- Voluntary after-the-fact donations – used by [shareware](#), [street performers](#), and [public broadcasting](#) in the United States.^[citation needed]
- Patron – For example, [open access](#) publishing relies on institutional and government funding of research faculty, who also have a professional incentive to publish for reputation and career advancement. Works of the U.S. federal government are automatically released into the [public domain](#).^[citation needed]
- **Freemium** – Give away a limited version for free and charge for a premium version (potentially using a [dual license](#)).
- Give away the product and charge something related – Charge for support of open-source [enterprise software](#), give away music but charge for concert admission.^[citation needed]



- Give away work in order to gain market share – Used by artists, in corporate software to spoil a dominant competitor (for example in the [browser wars](#) and the [Android operating system](#)).^[*citation needed*]
- For own use – Businesses or individual software developers often create software to solve a problem, bearing the full cost of initial creation. They will then open source the solution, and benefit from the improvements others make for their own needs. Communalizing the maintenance burden distributes the cost across more users; [free riders](#) can also benefit without undermining the creation process.

Open-source applications ^{[[edit](#)]}

Social and political views have been affected by the growth of the concept of open source. Advocates in one field often support the expansion of open source in other fields. But [Eric Raymond](#) and other founders of the [open-source movement](#) have sometimes publicly argued against speculation about applications outside software, saying that strong arguments for software openness should not be weakened by overreaching into areas where the story may be less compelling. The broader impact of the open-source movement, and the extent of its role in the development of new information sharing procedures, remain to be seen.



The [open-source movement](#) has inspired increased [transparency](#) and liberty in [biotechnology](#) research, for example by [open therapeutics](#)^[*?*] and [CAMBIA](#)^[*28*] Even the research methodologies themselves can benefit from the application of open-source principles.^[*29*] It has also given rise to the rapidly-expanding [open-source hardware](#) movement.

Computer software ^{[[edit](#)]}

Main article: [Free and open-source software](#)

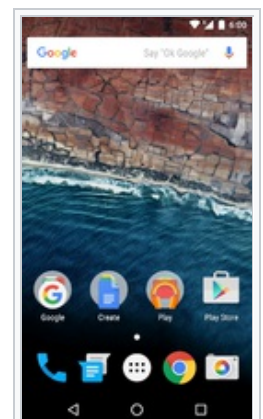
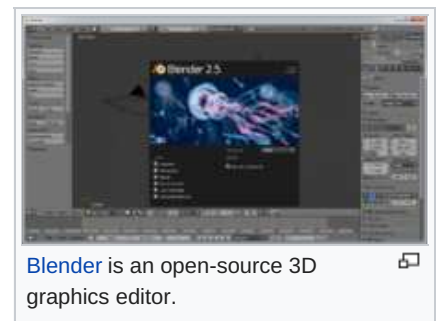
[Open-source software](#) is software which source code is published and made available to the public, enabling anyone to copy, modify and redistribute the source code without paying royalties or fees.^[*31*] Open-source code can evolve through community cooperation. These communities are composed of individual programmers as well as large companies. Some of the individual programmers who start an open-source project may end up establishing companies offering products or services incorporating open-source programs.^[*citation needed*]

Examples of open-source software products are:^[*32*]

- [Linux](#) (that much of world's server parks are running)
- [MediaWiki](#) (that Wikipedia is based upon)
- Many more:
 - [List of free and open-source software packages](#)
 - [List of formerly proprietary software](#)

Electronics ^{[[edit](#)]}

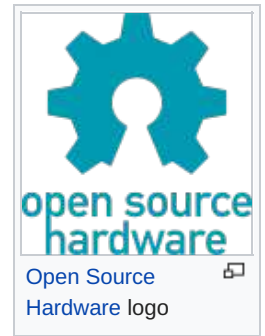
[Open-source hardware](#) is hardware which initial specification, usually in a software format, is published and made available to the public, enabling anyone to copy, modify and redistribute the hardware and source code without paying royalties or fees. Open-source hardware



[Android](#), the most popular mobile operating system (as of Nov 2012)^[*30*]

evolves through community cooperation. These communities are composed of individual hardware/software developers, hobbyists, as well as very large companies. Examples of open-source hardware initiatives are:

- **Openmoko**: a family of open-source [mobile phones](#), including the hardware specification and the [operating system](#).
- **OpenRISC**: an open-source microprocessor family, with architecture specification licensed under [GNU GPL](#) and implementation under [LGPL](#).
- **Sun Microsystems's OpenSPARC T1** Multicore processor. Sun has released it under [GPL](#).^[33]
- **Arduino**, a microcontroller platform for hobbyists, artists and designers.^[34]
- **GizmoSphere**, an open-source development platform for the embedded design community; the site includes code downloads and hardware schematics along with free user guides, spec sheets and other documentation.^[35]
- **Simputer**, an open hardware [handheld computer](#), designed in [India](#) for use in environments where computing devices such as personal computers are deemed inappropriate.^[36]
- **LEON**: A family of open-source microprocessors distributed in a library with peripheral [IP cores](#), open [SPARC V8](#) specification, implementation available under [GNU GPL](#).
- **Tinkerforge**: A system of open-source stackable microcontroller building blocks. Allows control of motors and read out sensors with the programming languages C, C++, C#, Object Pascal, Java, PHP, Python and Ruby over a USB or Wifi connection on Windows, Linux and Mac OS X. All of the hardware is licensed under [CERN OHL](#) (CERN Open Hardware License).
- **Open Compute Project**: designs for computer data center including power supply, Intel motherboard, AMD motherboard, chassis, racks, battery cabinet, and aspects of electrical and mechanical design.^[37]
- **Lasersaur**, an open-source [laser cutter](#).^[38]



Food and beverages [\[edit\]](#)

Some publishers of [open-access](#) journals have argued that [data](#) from [food science](#) and [gastronomy](#) studies should be freely available to aid [reproducibility](#).^[39] A number of people have published creative commons licensed recipe books.^[40]

- **Open-source colas** – cola soft drinks, similar to [Coca-Cola](#) and [Pepsi](#), whose recipe is open source and developed by volunteers. The taste is said to be comparable to that of the standard beverages. Most corporations producing beverages hold their formulas as closely guarded secrets.^[41]
- **Free Beer** (originally *Vores Øl*) – is an open-source beer created by students at the [IT-University](#) in [Copenhagen](#) together with [Superflex](#), an artist collective, to illustrate how open-source concepts might be applied outside the digital world.^{[42][43][44]}
- In 2002, the beer company [Brewtopia](#) in Australia started an open-source brewery and invited the general population to be involved in the development and ownership of the brewery, and to vote on the development of every aspect of its beer, Blowfly, and its road to market. In return for their feedback and input, individuals received shares in the company, which is now publicly traded on a stock exchange in Australia. The company has always



[Barack Obama](#) and [Dakota Meyer](#) drinking [White House Honey Ale](#) in 2011. The recipe is available for free.

adhered to its open-source roots and is the only beer company in the world that allows the public to design, customise and develop its own beers online.^[45]

Digital content [\[edit\]](#)

- Open-content projects organized by the Wikimedia Foundation – Sites such as Wikipedia and Wiktionary have embraced the open-content [Creative Commons](#) content licenses. These licenses were designed to adhere to principles similar to various open-source software development licenses. Many of these licenses ensure that content remains free for re-use, that source documents are made readily available to interested parties, and that changes to content are accepted easily back into the system. Important sites embracing open-source-like ideals are [Project Gutenberg](#)^[46] and [Wikisource](#), both of which post many books on which the copyright has expired and are thus in the [public domain](#), ensuring that anyone has free, unlimited access to that content.
- [Open ICEcat](#) is an open catalog for the IT, CE and Lighting sectors with product data-sheets based on [Open Content License](#) agreement. The digital content are distributed in XML and URL formats.
- [Google Sketchup's 3D Warehouse](#) is an open-source design community centered around the use of proprietary software that's free.
- The [University of Waterloo Stratford Campus](#) invites students every year to use its three-storey Christie [MicroTiles](#) wall as a digital canvas for their creative work.^[47]



Medicine [\[edit\]](#)

- Pharmaceuticals – There have been several proposals for open-source pharmaceutical development,^{[48][49]} which led to the establishment of the Tropical Disease Initiative^[50] and the Open Source Drug Discovery for Malaria Consortium.^[51]
- Genomics – The term "open-source genomics" refers to the combination of rapid release of sequence data (especially raw reads) and crowdsourced analyses from bioinformaticians around the world that characterised the analysis of the [2011 E. coli O104:H4 outbreak](#).^[52]
- [OpenEMR](#) – OpenEMR is an ONC-ATB Ambulatory EHR 2011-2012 certified electronic health records and medical practice management application. It features fully integrated electronic health, records, practice management, scheduling, electronic billing, and is the base for many EHR programs. <http://www.open-emr.org/>^[53]

Science and engineering [\[edit\]](#)

- Research – The [Science Commons](#) was created as an alternative to the expensive legal costs of sharing and reusing scientific works in journals etc.^[53]
- Research – [The Open Source Science Project](#) was created to increase the ability for students to participate in the research process by providing them access to microfunding – which, in turn, offers non-researchers the opportunity to directly invest, and follow, cutting-edge scientific research. All data and methodology is subsequently published in an openly accessible manner under a Creative Commons fair use license.
- Research – [The Open Solar Outdoors Test Field \(OSOTF\)](#)^[54] is a grid-connected [photovoltaic](#) test system, which continuously monitors the output of a number of photovoltaic modules and correlates their performance to a long list of highly accurate meteorological readings. The OSOTF is organized under open-source principles – All data and analysis is to be made freely available to the entire



photovoltaic community and the general public.^{[54][55]}

- Engineering – [Hyperloop](#), a form of high-speed transport proposed by entrepreneur [Elon Musk](#), which he describes as "an elevated, reduced-pressure tube that contains pressurized capsules driven within the tube by a number of linear electric motors".^[56]
- Construction – [WikiHouse](#) is an open-source project for designing and building [houses](#).^{[57][58]}
- Energy research - The [Open Energy Modelling Initiative](#) promotes open-source models and open data in energy research and policy advice.

Robotics [\[edit\]](#)

Main article: [Open-source robotics](#)

An open-source robot is a [robot](#) whose blueprints, schematics, or source code are released under an open-source model.

Transport [\[edit\]](#)

- Open Trip Planner – this code base is growing rapidly, with adoption in Portland, New York, The Netherlands and Helsinki.
- TravelSpirit – a greater level of 'super-architecture' ambition, to bring a range of open source projects together, in order to deliver 'Mobility as a Service'

Fashion [\[edit\]](#)

- [Eyewear](#) – In June 2013, an open-source eyewear brand, Botho, has started trading under the UK based Open Optics Ltd company.^[59]

Other [\[edit\]](#)

- Open-source principles can be applied to technical areas such as digital communication protocols and data storage formats.
- [Open-design](#) – which involves applying open-source methodologies to the design of artifacts and systems in the physical world. It is very nascent but has huge potential.^[60]
- [Open-source appropriate technology](#) (OSAT) refers to technologies that are designed in the same fashion as [free and open-source software](#).^[61] These technologies must be "[appropriate technology](#)" (AT) – meaning technology that is designed with special consideration to the environmental, ethical, cultural, social, political, and economic aspects of the community it is intended for. An example of this application is the use of open-source 3D printers like the [RepRap](#) to manufacture appropriate technology.^[62]
- [Teaching](#) – which involves applying the concepts of open source to instruction using a shared web space as a platform to improve upon learning, organizational, and management challenges. An example of an Open-source courseware is the Java Education & Development Initiative (JEDI).^[63] Other examples include [Khan Academy](#) and [wikiversity](#). At the university level, the use of [open-source-appropriate technology](#) classroom projects has been shown to be successful in forging the connection between science/engineering and social benefit.^[64] This approach has the potential to use university students' access to resources and testing equipment in furthering the development of [appropriate technology](#). Similarly OSAT has been used as a tool for improving [service learning](#).^{[65][66]}
- There are few examples of business information (methodologies, advice, guidance, practices) using the open-source model, although this is another case where the potential is enormous. [ITIL](#) is close to open source. It uses the [Cathedral model](#) (no mechanism exists for user contribution) and the content must be bought for a fee that is small by business consulting standards (hundreds of British pounds). Various checklists are published by government, banks or accounting firms.



- An open-source group emerged in 2012 that is attempting to design a firearm that may be downloaded from the internet and "printed" on a [3D Printer](#).^[67] Calling itself [Defense Distributed](#), the group wants to facilitate "a working plastic gun that could be downloaded and reproduced by anybody with a 3D printer".^[68]
- Agrecol, a German NGO has developed an open-source licence for seeds operating with [copyleft](#) and created OpenSourceSeeds as a respective service provider. Breeders that apply the license to their new invented material prevent it from the threat of privatisation and help to establish a commons-based breeding sector as an alternative to the commercial sector.^[69]
- [Open Source Ecology](#), farm equipment and global village construction kit.

Society and culture [edit]



This section's **factual accuracy is disputed**. Relevant discussion may be found on [Talk:Open-source model](#). Please help to ensure that disputed statements are [reliably sourced](#).
(December 2016) (*Learn how and when to remove this template message*)

The rise of open-source culture in the 20th century resulted from a growing tension between creative practices that involve require access to content that is often [copyrighted](#), and restrictive intellectual property laws and policies governing access to copyrighted content. The two main ways in which intellectual property laws became more restrictive in the 20th century were extensions to the term of copyright (particularly in the [United States](#)) and penalties, such as those articulated in the [Digital Millennium Copyright Act](#) (DMCA), placed on attempts to circumvent anti-piracy technologies.^[70]

Although artistic appropriation is often permitted under [fair-use](#) doctrines, the complexity and ambiguity of these doctrines creates an atmosphere of uncertainty among cultural practitioners. Also, the protective actions of copyright owners create what some call a "[chilling effect](#)" among cultural practitioners.^[71]

The idea of an "open-source" culture runs parallel to "[Free Culture](#)," but is substantively different. *Free culture* is a term derived from the [free software movement](#), and in contrast to that vision of culture, proponents of open-source culture (OSC) maintain that some intellectual property law needs to exist to protect cultural producers. Yet they propose a more nuanced position than corporations have traditionally sought. Instead of seeing intellectual property law as an expression of instrumental rules intended to uphold either natural rights or desirable outcomes, an argument for OSC takes into account diverse goods (as in "the Good life") and ends.

Sites such as [ccMixter](#) offer up free web space for anyone willing to license their work under a [Creative Commons](#) license. The resulting cultural product is then available to download free (generally accessible) to anyone with an Internet connection.^[72] Older analog technologies such as the [telephone](#) or [television](#) have limitations on the kind of interaction users can have.

Through various technologies such as [peer-to-peer](#) networks and [blogs](#), cultural producers can take advantage of vast [social networks](#) to distribute their products. As opposed to traditional media distribution, redistributing digital media on the Internet can be virtually costless. Technologies such as [BitTorrent](#) and [Gnutella](#) take advantage of various characteristics of the Internet protocol ([TCP/IP](#)) in an attempt to totally decentralize file distribution.

Further information: [Open-source software movement](#)

Government [edit]

- [Open politics](#) (sometimes known as *Open-source politics*) is a political process that uses Internet technologies such as blogs, email and polling to provide for a rapid feedback mechanism between political organizations and their supporters. There is also an alternative conception of the term *Open-source politics* which relates to the development of public policy

under a set of rules and processes similar to the open-source software movement.

- [Open-source governance](#) is similar to open-source politics, but it applies more to the democratic process and promotes the freedom of information.
- The [South Korean government](#) wants to increase its use of free and open-source software, in order to decrease its dependence on proprietary software solutions. It plans to make open standards a requirement, to allow the government to choose between multiple operating systems and web browsers. Korea's Ministry of Science, ICT & Future Planning is also preparing ten pilots on using open-source software distributions.^[73]

Ethics [\[edit\]](#)

Open-source ethics is split into two strands:

- Open-source ethics as an ethical school – Charles Ess and David Berry are researching whether ethics can learn anything from an open-source approach. Ess famously even defined the [AoIR Research Guidelines](#) as an example of open-source ethics.^[74]
- Open-source ethics as a professional body of rules – This is based principally on the computer ethics school, studying the questions of ethics and professionalism in the computer industry in general and software development in particular.^[75]

Religion [\[edit\]](#)

Irish philosopher Richard Kearney has used the term "open-source Hinduism" to refer to the way historical figures such as Mohandas Gandhi and Swami Vivekananda worked upon this ancient tradition.^[76]

Media [\[edit\]](#)

[Open-source journalism](#) formerly referred to the standard journalistic techniques of news gathering and fact checking, reflecting *open-source intelligence*, a similar term used in military intelligence circles. Now, *open-source journalism* commonly refers to forms of innovative publishing of [online journalism](#), rather than the sourcing of news stories by a professional journalist. In the 25 December 2006 issue of TIME magazine this is referred to as [user created content](#) and listed alongside more traditional open-source projects such as [OpenSolaris](#) and [Linux](#).

[Weblogs](#), or blogs, are another significant platform for open-source culture. Blogs consist of periodic, reverse chronologically ordered posts, using a technology that makes webpages easily updatable with no understanding of design, code, or [file transfer](#) required. While corporations, political campaigns and other formal institutions have begun using these tools to distribute information, many blogs are used by individuals for personal expression, political organizing, and socializing. Some, such as [LiveJournal](#) or [WordPress](#), utilize open-source software that is open to the public and can be modified by users to fit their own tastes. Whether the code is open or not, this format represents a nimble tool for people to borrow and re-present culture; whereas traditional websites made the illegal reproduction of culture difficult to regulate, the mutability of blogs makes "open sourcing" even more uncontrollable since it allows a larger portion of the population to replicate material more quickly in the public sphere.

[Messageboards](#) are another platform for open-source culture. Messageboards (also known as discussion boards or forums), are places online where people with similar interests can congregate and post messages for the community to read and respond to. Messageboards sometimes have moderators who enforce community standards of etiquette such as banning [spammers](#). Other common board features are private messages (where users can send messages to one another) as well as chat (a way to have a real time conversation online) and image uploading. Some messageboards use [phpBB](#), which is a free open-source package. Where blogs are more about individual expression and tend to revolve around their authors, messageboards are about creating a conversation amongst its users where information can be

shared freely and quickly. Messageboards are a way to remove intermediaries from everyday life—for instance, instead of relying on commercials and other forms of advertising, one can ask other users for frank reviews of a product, movie or CD. By removing the cultural middlemen, messageboards help speed the flow of information and exchange of ideas.

OpenDocument is an [open document file format](#) for saving and exchanging editable office documents such as text documents (including memos, reports, and books), [spreadsheets](#), charts, and presentations. Organizations and individuals that store their data in an open format such as OpenDocument avoid being [locked into](#) a single software vendor, leaving them free to switch software if their current vendor goes out of business, raises their prices, changes their software, or changes their [licensing](#) terms to something less favorable.

Open-source movie production is either an open call system in which a changing crew and cast collaborate in movie production, a system in which the end result is made available for re-use by others or in which exclusively open-source products are used in the production. The 2006 movie [Elephants Dream](#) is said to be the "world's first open movie",^[77] created entirely using [open-source technology](#).

An open-source documentary film has a production process allowing the open contributions of archival material [footage](#), and other filmic elements, both in unedited and edited form, similar to crowdsourcing. By doing so, on-line contributors become part of the process of creating the film, helping to influence the editorial and visual material to be used in the documentary, as well as its thematic development. The first open-source documentary film is the non-profit "[The American Revolution](#)", which went into development in 2006, and will examine the role media played in the cultural, social and political changes from 1968 to 1974 through the story of radio station WBCN-FM in Boston.^{[78][79][80][81]} The film is being produced by Lichtenstein Creative Media and the non-profit Filmmakers Collaborative. Open Source Cinema is a website to create Basement Tapes, a feature documentary about copyright in the digital age, co-produced by the National Film Board of Canada.^[82] **Open-source film-making** refers to a form of film-making that takes a method of idea formation from open-source software, but in this case the 'source' for a filmmaker is raw unedited footage rather than programming code. It can also refer to a method of film-making where the process of creation is 'open' i.e. a disparate group of contributors, at different times contribute to the final piece.

Open-IPTV is [IPTV](#) that is not limited to one recording studio, production studio, or cast. Open-IPTV uses the Internet or other means to pool efforts and resources together to create an online community that all contributes to a show.

Education ^[edit]

Within the academic community, there is discussion about expanding what could be called the "intellectual commons" (analogous to the [Creative Commons](#)). Proponents of this view have hailed the [Connexions](#) Project at [Rice University](#), [OpenCourseWare](#) project at [MIT](#), [Eugene Thacker's](#) article on "open-source DNA", the "Open Source Cultural Database", [Salman Khan's Khan Academy](#) and [Wikipedia](#) as examples of applying open source outside the realm of computer software.

Open-source curricula are instructional resources whose digital source can be freely used, distributed and modified.

Another strand to the academic community is in the area of research. Many funded research projects produce software as part of their work. There is an increasing interest in making the outputs of such projects available under an open-source license. In the UK the [Joint Information Systems Committee \(JISC\)](#) has developed a policy on open-source software. JISC also funds a development service called [OSS Watch](#) which acts as an advisory service for higher and further



education institutions wishing to use, contribute to and develop open-source software.

On 30 March 2010, President Barack Obama signed the Health Care and Education Reconciliation Act, which included \$2 billion over four years to fund the [TAACCCT program](#), which is described as "the largest OER (open education resources) initiative in the world and uniquely focused on creating curricula in partnership with industry for credentials in vocational industry sectors like manufacturing, health, energy, transportation, and IT".^[83]

Innovation communities [\[edit\]](#)

The principle of sharing pre-dates the open-source movement; for example, the free sharing of information has been institutionalized in the scientific enterprise since at least the 19th century. Open-source principles have always been part of the scientific community. The sociologist [Robert K. Merton](#) described the four basic elements of the community—universalism (an international perspective), communalism (sharing information), disinterestedness (removing one's personal views from the scientific inquiry) and organized skepticism (requirements of proof and review) that accurately describe the scientific community today.

These principles are, in part, complemented by US law's focus on protecting expression and method but not the ideas themselves. There is also a tradition of publishing research results to the scientific community instead of keeping all such knowledge proprietary. One of the recent initiatives in scientific publishing has been [open access](#)—the idea that research should be published in such a way that it is free and available to the public. There are currently many open access journals where the information is available free online, however most journals do charge a fee (either to users or libraries for access). The [Budapest Open Access Initiative](#) is an international effort with the goal of making all research articles available free on the Internet.

The [National Institutes of Health](#) has recently proposed a policy on "Enhanced Public Access to NIH Research Information". This policy would provide a free, searchable resource of NIH-funded results to the public and with other international repositories six months after its initial publication. The NIH's move is an important one because there is significant amount of public funding in scientific research. Many of the questions have yet to be answered—the balancing of profit vs. public access, and ensuring that desirable standards and incentives do not diminish with a shift to open access.

Farmavita.Net is a community of pharmaceutical executives that has recently proposed a new business model of open-source pharmaceuticals.^[84] The project is targeted to development and sharing of know-how for manufacture of essential and life-saving medicines. It is mainly dedicated to the countries with less developed economies where local pharmaceutical research and development resources are insufficient for national needs. It will be limited to generic (off-patent) medicines with established use. By definition, a medicinal product has a "well-established use" if it is used for at least 15 years, with recognized efficacy and an acceptable level of safety. In that event, the expensive clinical test and trial results could be replaced by appropriate scientific literature.


[Benjamin Franklin](#) was an early contributor eventually donating all his inventions including the [Franklin stove](#), [bifocals](#), and the [lightning rod](#) to the public domain.

New NGO communities are starting to use the open-source technology as a tool. One example is the Open Source Youth Network started in 2007 in Lisboa by ISCA members.^[85]

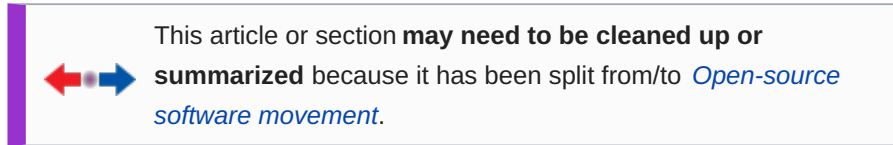
[Open innovation](#) is also a new emerging concept which advocates putting R&D in a common pool. The [Eclipse](#) platform is openly presenting itself as an Open innovation network.^[86]

Arts and recreation [\[edit\]](#)

Copyright protection is used in the performing arts and even in athletic activities. Some groups have attempted to remove copyright from such practices.^[87]

In 2012, Russian music composer, scientist and [Russian Pirate Party](#) member Victor Argonov presented detailed raw files of his electronic opera "2032"^[88] under free license [CC-BY-NC 3.0](#). This opera was originally composed and published in 2007 by Russian label [MC Entertainment](#) as a commercial product, but then the author changed its status to free. In his blog [\[2\]](#)  he said that he decided to open raw files (including wav, midi and other used formats) to the public in order to support worldwide pirate actions against [SOPA](#) and [PIPA](#). Several Internet resources,^{[89][90][91]} called "2032" the first open-source musical opera in history.

Other related movements [\[edit\]](#)



The following are events and applications that have been developed via the open source community, and echo the ideologies of the open source movement.^[92]

Open Education Consortium — an organization composed of various colleges that support open source and share some of their material online. This organization, headed by [Massachusetts Institute of Technology](#), was established to aid in the exchange of open source educational materials.

Wikipedia — user-generated [online encyclopedia](#) with sister projects in academic areas, such as [Wikiversity](#) — a community dedicated to the creation and exchange of learning materials^[93]*[not in citation given]*

Project Gutenberg — prior to the existence of [Google Scholar](#) Beta, this was the first supplier of [electronic books](#) and the very first free library project^[93]*[not in citation given]*

Synthetic Biology- This new technology is potentially important because it promises to enable cheap, lifesaving new drugs as well as helping to yield biofuels that may help to solve our energy problem. Although synthetic biology has not yet come out of its "lab" stage, it has potential to become industrialized in the near future. In order to industrialize open source science, there are some scientists who are trying to build their own brand of it.^[94]

Ideologically-related movements [\[edit\]](#)

The [open-access movement](#) is a movement that is similar in ideology to the open source movement. Members of this movement maintain that academic material should be readily available to provide help with “future research, assist in teaching and aid in academic purposes.” The Open access movement aims to eliminate subscription fees and licensing restrictions of academic materials^[95]

The [free-culture movement](#) is a movement that seeks to achieve a culture that engages in collective freedom via freedom of expression, free public access to knowledge and information, full demonstration of creativity and innovation in various arenas and promotion of citizen liberties.^[96]*[citation needed]*

[Creative Commons](#) is an organization that “develops, supports, and stewards legal and technical infrastructure that maximizes digital creativity, sharing, and innovation.” It encourages the use of protected properties online for research, education, and creative purposes in pursuit of a universal access. Creative Commons provides an infrastructure through a set of copyright licenses and tools that creates a better balance within the realm of “all rights reserved” properties.^[97] The Creative Commons license offers a slightly more lenient alternative to “all rights reserved” copyrights for those who do not wish to exclude the use of their material.^[98]

The [Zeitgeist Movement](#) is an international social movement that advocates a transition into a [sustainable](#) "resource-based economy" based on [collaboration](#) in which monetary incentives are replaced by commons-based ones with everyone [having access](#) to everything (from code to

products) as in "open source everything".^{[99][100]} While its activism and events are typically focused on media and education, TZM is a major supporter of open source projects worldwide since they allow for uninhibited advancement of science and technology, independent of constraints posed by institutions of patenting and capitalist investment.^[101]

P2P Foundation is an "international organization focused on studying, researching, documenting and promoting **peer to peer practices** in a very broad sense". Its objectives incorporate those of the open source movement, whose principles are integrated in a larger socio-economic model.^[102]

See also [edit]



This "see also" section **may contain an excessive number of suggestions**. Please ensure that only the most relevant links are given, that they are not **red links**, and that any links are not already in this article. (June 2015) (*Learn how and when to remove this template message*)

Lists [edit]

- [List of commercial open-source applications](#)
- [List of open-source healthcare software](#)
- [List of open-source software packages](#)
- [List of open-source video games](#)
- [List of trademarked open-source software](#)
- [List of open-source Android applications](#)



Terms based on open source [edit]

- [Open-source appropriate technology](#)
- [Open-source economics](#)
- [Open Source Ecology](#)
- [Open-source governance](#)
- [Open-source hardware](#)
- [Open Source Initiative](#)
- [Open-source license](#)
- [Open-source political campaign](#)
- [Open-source record label](#)
- [Open-source religion](#)
- [Open-source robotics](#)
- [Open-source software](#)
- [Open-source movement](#)

Other [edit]

- *Open Sources: Voices from the Open Source Revolution* (book)
- [Business models for open-source software](#)
- [Collaborative intelligence](#)
- [Commons-based peer production](#)
- [Commercial open-source applications](#)
- [Community source](#)
- [Digital freedom](#)
- [Diseconomy of scale](#)
- [Embrace, extend and extinguish](#)
- [Free Beer](#)
- [Free software](#)
- [Gift economy](#)
- [Glossary of legal terms in technology](#)






- [Halloween Documents](#)
- [Linux](#)
- [Mass collaboration](#)
- [Network effect](#)
- [Open access \(publishing\)](#)
- [Open content](#)
- [Open data](#)
- [Open-design movement](#)
- [Open format](#)
- [Open implementation](#)
- [Open innovation](#)
- [OpenJDK](#)
- [Open research](#)
- [Open security](#)
- [OpenSolaris](#)
- [Open Source Ecology](#)
- [Open Source Lab \(book\)](#)
- [Comparison of open source and closed source](#)
- [Open system \(computing\)](#)
- [Open standard](#)
- [OpenDWG](#)
- [Openness](#)
- [Peer production](#)
- [Proprietary software](#)
- [Shared source](#)
- [Sharing economy](#)
- [Vendor lock-in](#)
- [Web literacy \(Open Practices\)](#)

References [[edit](#)]

1. [^] Levine, Sheen S., & Prietula, M. J. (2013). [Open Collaboration for Innovation: Principles and Performance](#). *Organization Science*, doi:10.1287/orsc.2013.0872
2. [^] Raymond, Eric S. (2001). *The cathedral and the bazaar: musings on Linux and Open Source by an accidental revolutionary*. OReilly. ISBN 978-0-596-00108-7.^[*page needed*]
3. [^] Pearce, Joshua M (2012). "The Case for Open Source Appropriate Technology". *Environment, Development and Sustainability*. **14** (3): 425–431. doi:10.1007/s10668-012-9337-9.
4. [^] "Science 2.0 is here as CSIR resorts to open-source drug research for TB" *Business Standard*, 1 March 2009
5. [^] "Open Source Drug Discovery for Malaria Consortium"
6. [^] Lakhani, K.R.; von Hippel, E. (June 2003). "How Open Source Software Works: Free User to User Assistance". *Research Policy*. **32** (6): 923–943. doi:10.1016/S0048-7333(02)00095-1. hdl:1721.1/70028.
7. [^] Gerber, A.; Molefo, O.; Van der Merwe, A. (2010). "Documenting open-source migration processes for re-use". In Kotze, P.; Gerber, A.; van der Merwe, A.; et al. *Proceedings of the SAICSIT 2010 Conference — Fountains of Computing Research*. ACM Press. pp. 75–85. CiteSeerX [10.1.1.1033.7791](#) . doi:10.1145/1899503.1899512. ISBN 978-1-60558-950-3.
8. [^] Weber 2004^[*page needed*]
9. [^] [a](#) [b](#) [c](#) [d](#) James J. Flink (1977). *The Car Culture*. MIT Press. ISBN 978-0-262-56015-3.
10. [^] Fisher, Franklin M.; James W. McKie; Richard B. Mancke (1983). *IBM and the U.S. Data Processing Industry: An Economic History*. Praeger. pp. 172–9. ISBN 978-0-03-063059-0. IBM unbundled (began charging for) software 23 June 1969
11. [^] [Dave Pitts' IBM 7090 support Archived 2015-08-27 at the Wayback Machine](#) – An example of distributed source: Page contains a link to [IBM 7090/94 IBSYS](#) source, including [COBOL](#) and [FORTRAN](#) compilers.













12. ^ O'Mahony, Siobhan Clare (2002). "The emergence of a new commercial actor: Community managed software projects". Stanford, CA: Stanford University: 34–42.
13. ^ Eric S. Raymond. "Goodbye, "free software"; hello, "open source" ". "The problem with it is twofold. First, ... the term "free" is very ambiguous ... Second, the term makes a lot of corporate types nervous."
14. ^ Shea, Tom (1983-06-23). "Free software - Free software is a junkyard of software spare parts". *InfoWorld*. Retrieved 2016-02-10. ""In contrast to commercial software is a large and growing body of free software that exists in the public domain. Public-domain software is written by microcomputer hobbyists (also known as "hackers") many of whom are professional programmers in their work life. [...] Since everybody has access to source code, many routines have not only been used but dramatically improved by other programmers.""
15. ^ a b c Tiemann, Michael (19 September 2006). "History of the OSI" . Open Source Initiative . Archived from the original on 1 October 2002. Retrieved 23 August 2008.
16. ^ "Why Open Source misses the point of Free Software" . fsf.org. 2012-05-18. Retrieved 2012-11-14.
17. ^ Muffatto, Moreno (2006). *Open Source: A Multidisciplinary Approach* . Imperial College Press. ISBN 978-1-86094-665-3.
18. ^ "Goodbye, "free software"; hello, "open source" ". Catb.org. Retrieved 2012-10-25.
19. ^ a b van Rossum, Guido (1998-04-10). "Open Source Summit" . *Linux Gazette*. Archived from the original on 29 December 2013. Retrieved 2015-02-07.
20. ^ a b "Open Source Studie Schweiz 2015 publiziert: Mehr Unabhängigkeit und Sicherheit dank Open Source" . *Der Verband für den Informatiker & für die Informatik* (in German). Retrieved 2018-12-23.
21. ^ Granstrand, Ove (1999). *The economics and management of intellectual property : towards intellectual capitalism*. Cheltenham, UK: E. Elgar. ISBN 978-1-85898-967-9.
22. ^ "About The Licenses" . Creative Commons. Retrieved 2012-10-25.
23. ^ J.M. Pearce, *Open-Source Lab: How to Build Your Own Hardware and Reduce Research Costs* Elsevier, 2014
24. ^ 3D Printed Open Hardware Syringe Yields \$800M Value, Study Finds - 3D Printing Industry 2-2-2015
25. ^ Wittbrodt, B.T.; Glover, A.G.; Laureto, J.; Anzalone, G.C.; Oppliger, D.; Irwin, J.L.; Pearce, J.M. (2013). "Life-cycle economic analysis of distributed manufacturing with open-source 3-D printers". *Mechatronics*. **23** (6): 713–726. doi:10.1016/j.mechatronics.2013.06.002.
26. ^ Joe McCann, "The Meteoric Rise Of Open Source And Why Investors Should Care" , *Forbes* (in German), retrieved 2017-10-10
27. ^ Dharmesh Thakker, "Tracking the explosive growth of open-source software" , *TechCrunch* (in German), retrieved 2017-10-10
28. ^ "'Open-Source Practices for Biotechnology' — Cambia — Enabling Innovation" . Cambia. Retrieved 2012-10-25.
29. ^ Pearce, Joshua M. (1 August 2012). "Open Source Research in Sustainability". *Sustainability: The Journal of Record*. **5** (4): 238–243. doi:10.1089/sus.2012.9944.
30. ^ "Android Marks Fourth Anniversary Since Launch with 75.0% Market Share in Third Quarter, According to IDC — prUS23771812" . Idc.com. Archived from the original on 3 November 2012. Retrieved 3 November 2012.
31. ^ "FAQ | LucidWorks" . Lucidimagination.com. 2012-08-08. Retrieved 2012-10-25.
32. ^ Randolph Metcalfe (2012-01-31). "Examples of Open Source Software" . OSS Watch. Retrieved 2012-10-25.
33. ^ "An Open Source Processor used in Sun SPARC Servers" . OpenSPARC. Retrieved 2012-10-25.
34. ^ "Arduino — HomePage" . Arduino.cc. Retrieved 2012-10-25.
35. ^ "GizmoSphere Website" . gizmosphere.org. 2013.
36. ^ "Computers: Systems: Handhelds: Open Source: Simputer" . *Open Directory*. Dmoz.org. Retrieved 2012-10-25.
37. ^ "Download Specs & Designs" . *Open Compute Project website* . Facebook Inc. 2011. Retrieved 2011-08-08.
38. ^ "Lasersaur — HomePage" . Lasersaur.com. Retrieved 2013-05-25.
39. ^ <http://bloas.biomedcentral.com/bmcbloa/2013/12/02/meda-bites-how-open-data-and->

- gastronomy-are-proving-a-recipe-for-success/
40. ^ Martin Lersch. "TEXTURE - A hydrocolloid recipe collection" (PDF).
 41. ^ "Open Cola Soft Drink Formula, Version 1.1.3 from Tyromaniac" (PDF). Alfredo.octavio.net. Retrieved 2012-10-25.
 42. ^ The concept expands upon a statement found in the [Free Software Definition](#): "Free software is a matter of liberty, not price. To understand the concept, you should think of 'free' as in 'free speech' not as in 'free beer.'"
 43. ^ Stallman, Richard M. "The Gnu Organisation: What is free software?" Gnu.org. Retrieved 2012-10-25.
 44. ^ Cohn, David (18 July 2005). "Free Beer for Geeks". *Wired News*.
 45. ^ "About Brewtopia". *Brewtopia*. Brewtopia.com. Retrieved 2017-07-16.
 46. ^ "Project Gutenberg — free ebooks". Gutenberg.org. Retrieved 2012-10-25.
 47. ^ "Want to take 3 storeys to tell your story". uwaterloo.ca/stratford-campus. Retrieved 2013-04-07.
 48. ^ Munos B (September 2006). "Can open-source R&D reinvigorate drug research?". *Nat Rev Drug Discov*. **5** (9): 723–9. doi:10.1038/nrd2131. PMID 16915233.
 49. ^ Maurer SM, Rai A, Sali A (December 2004). "Finding cures for tropical diseases: is open source an answer?". *PLoS Med*. **1** (3): e56. doi:10.1371/journal.pmed.0010056. PMC 539047. PMID 15630466.
 50. ^ "the Tropical Disease Initiative". Tropicaldisease.org. 2009-07-16. Retrieved 2012-10-25.
 51. ^ "The Open Source Drug Discovery for Malaria Consortium". Openwetware.org. 2012-12-14. Retrieved 2013-03-18.
 52. ^ Rohde H; Qin, J; Cui, Y; Li, D; Loman, NJ; Hentschke, M; Chen, W; Pu, F; et al. (2011). "Open-Source Genomic Analysis of Shiga-Toxin–Producing *E. coli* O104:H4". *N Engl J Med*. **365** (8): 718–24. doi:10.1056/NEJMoa1107643. PMID 21793736.
 53. ^ "Science — Creative Commons". Sciencecommons.org. 2010-11-17. Retrieved 2012-10-25.
 54. ^ ^a ^b Pearce, Joshua. M; Adegboyega Babasola; Rob Andrews (2012). "Open Solar Photovoltaic Systems Optimization" (PDF). *Proceedings of the 16th Annual National Collegiate Inventors and Innovators Alliance Conference*: 1–7.
 55. ^ Buitenhuis, A.J; Pearce, J.M (2012). "Open Solar Outdoors Test Field — Appropedia: The sustainability wiki". *Energy for Sustainable Development*. **16** (3): 379–388. doi:10.1016/j.esd.2012.06.006. Retrieved 2012-10-25.
 56. ^ Musk, Elon (12 August 2013). "Hyperloop". *Tesla Motors website*. Retrieved 22 September 2013.
 57. ^ LaBarre, Suzanne (25 August 2011). "WikiHouse, An Online Building Kit, Shows How To Make A House In 24 Hours". *Co.Design*. Fast Company, Inc. Retrieved 17 December 2013.
 58. ^ Kingsley, Jeremy (22 February 2012). "The WikiHouse Revolution". *Slate*. The Slate Group, LLC. Retrieved 17 December 2013.
 59. ^ "Botho — The Open Source Eyewear Brand". www.botho.co.uk. Retrieved 2012-06-24.
 60. ^ "Open collaborative design". AdCiv. 2010-07-29. Retrieved 2012-10-25.
 61. ^ A. J. Buitenhuis, I. Zelenika and J. M. Pearce, "Open Design-Based Strategies to Enhance Appropriate Technology Development", *Proceedings of the 14th Annual National Collegiate Inventors and Innovators Alliance Conference : Open*, 25–27 March 2010, pp. 1-12.
 62. ^ J.M. Pearce; C. Morris Blair; K.J. Laciak; R. Andrews; A. Nosrat; I. Zelenika-Zovko (2010). "3-D Printing of Open Source Appropriate Technologies for Self-Directed Sustainable Development". *Journal of Sustainable Development*. **3** (4): 17–29. doi:10.5539/jsd.v3n4p17.
 63. ^ "JEDI: The Open Source Curriculum — Project Kenai". Kenai.com. Retrieved 2012-10-25.
 64. ^ J.M. Pearce (2007). "Teaching Physics Using Appropriate Technology Projects". *The Physics Teacher*. **45** (3): 164–7. Bibcode:2007PhTea..45..164P. doi:10.1119/1.2709675.
 65. ^ Joshua M. Pearce (2009). "Appropedia as a Tool for Service Learning in Sustainable Development". *Journal of Education for Sustainable Development*. **3** (1): 45–53. CiteSeerX 10.1.1.620.246. doi:10.1177/097340820900300112. Q-Space pre-print
 66. ^ S. Murphy and N. Saleh, "Information literacy in CEAB's accreditation criteria: the hidden attribute", In *Proceedings of the Sixth International Conference on Innovation and Practices in Engineering Design and Engineering Education*, 2009. Hamilton, ON 27–29 July 2009.
 67. ^ Greenberg, Andy (2012-08-23). "'Wiki Weapon Project' Aims To Create A Gun Anyone Can 3D-

- [Print At Home](#)⁴. *Forbes*. Retrieved 2012-08-27.
68. [^] Poeter, Damon (2012-08-24). "Could a 'Printable Gun' Change the World?" ⁴. *PC Magazine*. Retrieved 2012-08-27.
 69. [^] "New Open Source License Fights the Enclosure of Seeds | David Bollier" ⁴. www.bollier.org. Retrieved 2017-07-03.
 70. [^] US Copyright Office (December 1998). "The Digital Millennium Copyright Act of 1998"  (PDF). Retrieved 2012-10-25.
 71. [^] "Chilling Effects Clearinghouse" ⁴. Chillingeffects.org. Retrieved 2012-10-25.
 72. [^] "Open source software development – Just another case of collective invention?" ⁴. Archived from the original ⁴ on 2011-12-22.
 73. [^] <https://joinup.ec.europa.eu/community/osor/news/south-korean-government-switch-open-source>⁴
 74. [^] Berry (2004) Internet Ethics: Privacy, Ethics and Alienation – An Open Source Approach.  (PDF file)
 75. [^] El-Emam, K (2001). "Ethics and Open Source" ⁴. *Empirical Software Engineering*. **6** (4).
 76. [^] Kearney, Richard (2011). *Anatheism: Returning to God After God* ⁴. Columbia University Press. p. xiii. ISBN 978-0-231-14789-7.
 77. [^] "Elephants Dream" ⁴. Elephants Dream. 2006-08-13. Retrieved 2012-10-25.
 78. [^] "Web Power to the People," The Boston Herald, 29 September 2005
 79. [^] "The Glory Days of the Rock of Boston," Boston Globe, Op-Ed, 18 July 2009.
 80. [^] "'BCN Documentary Looks for Archival Contributors," Boston Herald, 11 August 2009
 81. [^] "WBCN and The American Revolution," The Huffington Post, 11 August 2009
 82. [^] "Watch Documentaries and Animated Films Online" ⁴. NFB.ca. 2012-08-27. Retrieved 2012-10-25.
 83. [^] http://openeducation2013.sched.org/event/23553161b01967fd7ff58eea22ce24b2#.U_qxw2Mpf8⁴
 84. [^] "Pharmaceutical Licensing Network — Open Source Pharmaceuticals — New Business Model" ⁴. Farmavita.Net. 2007-01-12. Retrieved 2012-10-28.
 85. [^] [1] ⁴ Archived ⁴ 19 March 2007 at the [Wayback Machine](#)
 86. [^] "Eclipse"  (PDF). Eclipse.org. Retrieved 2012-10-25.
 87. [^] "Open Source Yoga Unity — Home" ⁴. Yogaunity.org. Retrieved 2012-10-25.
 88. [^] "Техно-опера "2032: Легенда о несбывшемся грядущем" " ⁴. Argonov.ru. Retrieved 2013-06-07.
 89. [^] "Первая в мире музыкальная опера с открытыми исходниками — 25 Января 2012 — Новости музыки и шоубизнеса" ⁴. Muznews.rockzvuk.com. Retrieved 2013-06-07.
 90. [^] "Музыкальная опера "2032" становится opensource — Talks — Форум" ⁴. Linux.org.ru. Retrieved 2013-06-07.
 91. [^] "Первая в мире музыкальная опера с открытыми исходниками" ⁴. Netskop.ru. Retrieved 2013-06-07.
 92. [^] "The case for open source: Open source has made significant leaps in recent years. what does it have to offer education?" ⁴. *Technology & Learning*. **27** (7): 16. 2007.
 93. [^] ^a ^b Warger, T. (2002). *The Open Source Movement*  Archived  17 July 2011 at the [Wayback Machine](#). Retrieved 22 November 2009, from Education Resources Information Center
 94. [^] Wilson Center. (2009). *Synthetic Biology: Feasibility of the Open Source Movement* ⁴. Wislson On Demand Center.
 95. [^] Harnad, S. (14 November 2009). *Zine & Articles: Open Access Movement and Its Implications for the Future of Academic Writing* ⁴. Retrieved 22 November 2009.
 96. [^] Students For Free Culture. (2009). *Main Page* ⁴. Retrieved 22 November 2009, from free culture.org
 97. [^] "Creative Commons Organization" ⁴. Creative Commons. Retrieved 2011-10-20.
 98. [^] "How to find Creative Commons images on Flickr" ⁴. *New Media Rights*. Retrieved 2011-10-20.
 99. [^] Dostaler, Kim (19 July 2012). "The Zeitgeist Movement: Getting to the Root of the Problem" ⁴. Retrieved 10 June 2015.
 100. [^] Dobson, Stuart (27 January 2013). "Forget the Class War, How to Win the Fight Against the Elite" ⁴. Archived from the original ⁴ on 11 June 2015. Retrieved 10 June 2015.
 101. [^] "Open Source Ecology - Zeitgeist movement" ⁴. Open Source Ecology.

102. ^ "P2P Foundation:About" [P2P Foundation](#) . Retrieved 2015-02-15.

Further reading [\[edit\]](#)

- Benkler, Yochai (2006). *The Wealth of Networks: How Social Production Transforms Markets and Freedom*  (PDF). Yale University Press.
- Berry, David M. (2008). *Copy, Rip, Burn: The Politics of Copyleft and Open Source*. London:Pluto Press. ISBN 978-0745324142.
- Karl Fogel. [Producing Open Source Software](#)  (How to run a successful free-software project). Free PDF version available.
- Goldman, Ron; Gabriel, Richard P. (2005). *Innovation Happens Elsewhere: Open Source as Business Strategy* . Richard P. Gabriel. ISBN 978-1-55860-889-4.
- Dunlap, Isaac Hunter (2006). *Open Source Database Driven Web Development: A Guide for Information Professionals* . Oxford: Chandos. ISBN 978-1-84334-161-1.
- Kostakis, V.; Bauwens, M. (2014). *Network Society and Future Scenarios for a Collaborative Economy*. Palgrave Macmillan. ISBN 978-1-137-41506-6. [\(wiki\)](#) 
- Nettingsmeier, Jörn. "So What? I Don't Hack!" *eContact! 11.3 – Logiciels audio " open source " / Open Source for Audio Application*  (September 2009). Montréal: CEC.
- Stallman, Richard M. *Free Software Free Society: Selected essays of Richard M. Stallman* .
- Schrape, Jan-Felix (2017). "Open-source projects as incubators of innovation. From niche phenomenon to integral part of the industry". *Convergence*: 135485651773579. doi:10.1177/1354856517735795 .
- Various authors. *eContact! 11.3 – Logiciels audio " open source " / Open Source for Audio Application*  (September 2009). Montréal: CEC.
- Various authors. "Open Source Travel Guide [\[wiki\]](#) ". *eContact! 11.3 – Logiciels audio " open source " / Open Source for Audio Application*  (September 2009). Montréal: CEC.
- Weber, Steve (2004). *The Success of Open Source* . Harvard University Press. ISBN 978-0-674-01292-9.
- Ray, Partha Pratim; Rai, Rebika (2013). *Open Source Hardware: An Introductory Approach*. Lap Lambert Publishing House. ISBN 978-3-659-46591-8.

Literature on legal and economic aspects [\[edit\]](#)

- Benkler, Y. (December 2002). "Coase's Penguin, or, Linux and The Nature of the Firm"  (PDF). *Yale Law Journal*. **112** (3): 369–446. arXiv:cs/0109077 . doi:10.2307/1562247 . hdl:10535/2974 . JSTOR 1562247 .
- Berry, D.M.; Moss, G. (2008). "Libre Culture: Meditations on Free Culture"  (PDF). Canada: Pygmalion Books.
- Bitzer, J.; Schröder, P.J.H. (2005). "The Impact of Entry and Competition by Open Source Software on Innovation Activity"  (PDF). *Industrial Organization 0512001*. EconWPA.
- v. Engelhardt, S. (2008). "The Economic Properties of Software"  (PDF). *Jena Economic Research Papers*. **2**: 2008–045.
- v. Engelhardt, S. (2008): "Intellectual Property Rights and Ex-Post Transaction Costs: the


Case of Open and Closed Source Software", Jena Economic Research Papers 2008-047 [↗](#). (PDF)

- v. Engelhardt, S.; Swaminathan, S. (2008). "Open Source Software, Closed Source Software or Both: Impacts on Industry Growth and the Role of Intellectual Property Rights" [↗](#) (PDF). *Discussion Papers of DIW Berlin* 799.
- European Commission. (2006). *Economic impact of open source software on innovation and the competitiveness of the Information and Communication Technologies sector in the EU* [↗](#). Brussels.
- Feller, J.; Fitzgerald, B.; Hissam, S.A., eds. (2005). *Perspectives on Free and Open Source Software* [↗](#). MIT Press. ISBN 978-0-262-06246-6.
- v. Hippel, E.; v. Krogh, G. (2003). "Open source software and the "private-collective" innovation model: Issues for organization science". *Organization Science*. **14** (2): 209–223. doi:10.1287/orsc.14.2.209.14992 [↗](#). hdl:1721.1/66145 [↗](#).
- Kostakis, V.; Bauwens, M. (2014). *Network Society and Future Scenarios for a Collaborative Economy*. Palgrave Macmillan. ISBN 978-1-137-41506-6. (wiki) [↗](#)
- Lerner J., Pathak P. A., Tirole, J. (2006). "The Dynamics of Open Source Contributors" [↗](#). *American Economic Review*. **96** (2): 114–8. CiteSeerX [↗](#) 10.1.1.510.9948 . doi:10.1257/000282806777211874 [↗](#).
- Lerner, J., Tirole, J. (2002). "Some simple economics on open source". *Journal of Industrial Economics*. **50** (2): 197–234. CiteSeerX [↗](#) 10.1.1.461.3373 . doi:10.1111/1467-6451.00174 [↗](#). earlier revision (PDF) [↗](#)
- Lerner, J.; Tirole, J. (2005). "The Scope of Open Source Licensing". *The Journal of Law, Economics, and Organization*. **21**: 20–56. CiteSeerX [↗](#) 10.1.1.72.465 . doi:10.1093/jleo/ewi002 [↗](#).
- Lerner, J.; Tirole, J. (2005). "The Economics of Technology Sharing: Open Source and Beyond". *Journal of Economic Perspectives*. **19** (2): 99–120. doi:10.1257/0895330054048678 [↗](#).
- Lerner, J.; Tirole, J. (2005). "The Economics of Technology Sharing: Open Source and Beyond". *Journal of Economic Perspectives*. **19** (2): 99–120. doi:10.1257/0895330054048678 [↗](#).
- Maurer, S.M. (2008). "Open source biology: Finding a niche (or maybe several)". *UMKC Law Review*. **76** (2). doi:10.2139/ssrn.1114371 [↗](#). SSRN [↗](#) 1114371 .
- Osterloh, M.; Rota, S. (2007). "Open source software development — Just another case of collective invention?" [↗](#) (PDF). *Research Policy*. **36** (2): 157–171. doi:10.1016/j.respol.2006.10.004 [↗](#).
- Riehle, D. (April 2007). "The Economic Motivation of Open Source: Stakeholder Perspectives" [↗](#). *IEEE Computer*. **40** (4): 25–32. doi:10.1109/MC.2007.147 [↗](#).
- Rossi, M.A. (2006). "Decoding the free/open source software puzzle: A survey of theoretical and empirical contributions" [↗](#) (PDF). In Bitzer, J.; Schröder, P. *The Economics of Open Source Software Development*. Elsevier. pp. 15–55. ISBN 978-0-444-52769-1.
- Schiff, A. (2002). "The Economics of Open Source Software: A Survey of the Early Literature" [↗](#) (PDF). *Review of Network Economics*. **1** (1): 66–74. doi:10.2202/1446-9022.1004 [↗](#).
- Schwarz, M.; Takhteyev, Y. (2010). "Half a Century of Public Software Institutions: Open Source as a Solution to the Hold-Up Problem". *Journal of Public Economic Theory*. **12** (4): 609–639. CiteSeerX [↗](#) 10.1.1.625.2368 . doi:10.1111/j.1467-9779.2010.01467.x [↗](#). earlier revision [↗](#)
- Spagnoletti, P.; Federici, T. (2011). "Exploring the Interplay Between FLOSS Adoption and Organizational Innovation" [↗](#). *Communications of the Association for Information Systems (CAIS)*. **29** (15): 279–298.
- Abramson, Bruce (2005). *Digital Phoenix; Why the Information Economy Collapsed and How it Will Rise Again*. MIT Press. ISBN 978-0-262-51196-4.


- Sampathkumar, K.S. *Understanding FOSS Version 4.0 revised*[↗]. ISBN 978-8-184-65469-1.

External links ^[edit]






- [Open Source Malaria](#)[↗]
- [Open Source Pharma](#)[↗]
- [Open Source Tuberculosis](#)[↗]
- [What is open source?](#)[↗] (opensource.com)
- "An open-source shot in the arm?"[↗] *The Economist*, Jun 10th 2004
- [Google-O'Reilly Open Source Awards](#)[↗]
- [UNU/IIST Open Source Software Certification](#)[↗]
- [Open Source Open World – Open Standards Throughout the Globe](#)[↗]
- *The Changelog*[↗], a podcast and blog that covers what's fresh and new in Open Source (essentially covering "the changelog" of open source projects)
- [Can We Open Source Everything? The Future of the Open Philosophy](#). University of Cambridge.[↗]



Wikibooks has a book on the topic of: *[Open Source](#)*



Look up *[open source](#)* in Wiktionary, the free dictionary.

V · T · E Free and open-source software	
General	Alternative terms for free software · Comparison of open-source and closed-source software · Comparison of source code hosting facilities · Free software · Free software project directories · Gratis versus libre · Long-term support · Open-source software · Open-source software development · Outline
Software packages	Audio · Bioinformatics · Codecs · Collaboration · Configuration management · Device drivers (Graphics · Wireless) · Geophysics · Health · Mathematics · Operating systems · Programming languages · Routing · Statistics · Television · Video games · Web applications (Content management systems · E-commerce) · Word processors · Android apps · iOS apps · Commercial · Trademarked · Formerly proprietary
Community	Free software movement · History · Open-source software movement · Organizations · Events
Licenses	AFL · Apache · APSL · Artistic · Beerware · Boost · BSD · CC0 · CDDL · EPL · Free Software Foundation (GNU GPL · GNU LGPL) · ISC · MIT · MPL · Ms-PL/RL · Python · Python Software Foundation License · Sleepycat · Unlicense · WTFPL · zlib <div style="margin-left: 20px;"> Types and standards <ul style="list-style-type: none"> Comparison of free and open-source software licenses · Contributor License Agreement · Copyleft · Debian Free Software Guidelines · Definition of Free Cultural Works · Free license · The Free Software Definition · The Open Source Definition · Open-source license · Permissive software licence · Public domain · Viral license </div>
Challenges	Binary blob · Digital rights management · Hardware restrictions · License proliferation · Mozilla software rebranding · Proprietary software · SCO/Linux controversies · Secure boot · Software patents · Software security · Trusted Computing
Related topics	Forking · <i>GNU Manifesto</i> · Microsoft Open Specification Promise · Open-core model · Open-source hardware · Shared Source Initiative · Source-available software · <i>The Cathedral and the Bazaar</i> · <i>Revolution OS</i>
 Book ·  Category ·  Commons ·  Portal ·  WikiProject	

V · T · E Intellectual property activism	
Issues	Artificial scarcity · Copyright infringement · Digital rights management · Gripe site · Legal aspects of file sharing · Mashup (digital · music · videos) · Monopolies of knowledge · Music piracy · Orphan works · Patents (biological · software · software patent debate · trolling) · Public domain
Concepts	All rights reversed · Alternative compensation system · Anti-copyright notice · Business models for open-source software · Copyleft · Commercial use of copyleft works · Commons-based peer production · Free content · Free software license · Libertarian positions · Open content · Open-design movement · Open Music Model ·

	Open patent · Open-source hardware · Open-source software · Prize system (contests) · Share-alike · Video on demand
Movements	Access to Knowledge movement · Anti-copyright · Cultural environmentalism · Free-culture movement · Free software movement
Organizations	Pro-copyright Copyright Alliance
	Pro-copyleft Creative Commons · Electronic Frontier Foundation · Free Software Foundation · Open Rights Group · Organization for Transformative Works · The Pirate Bay · Piratbyrå · Pirate Party · Sci-Hub · Students for Free Culture
People	Alexandra Elbakyan · Rick Falkvinge · Lawrence Lessig · Richard Stallman · Peter Sunde · Peter Suber · Aaron Swartz
Documentaries	<i>Steal This Film</i> (2006, 2007) · <i>Good Copy Bad Copy</i> (2007) · <i>RiP!: A Remix Manifesto</i> (2008) · <i>TPB AFK: The Pirate Bay Away From Keyboard</i> (2013) · <i>The Internet's Own Boy</i> (2014)

V · T · E		Free-culture and open-source movements		
Concepts and practices	Key concepts	Commons-based peer production · Crowdsourcing · <i>Gratis</i> versus <i>libre</i> · Open collaboration · Openness · Participatory culture · Sharism	Social peer-to-peer processes Peer-to-peer carsharing · Peer-to-peer lending · Peer-to-peer property rental · Peer-to-peer renting · Peer-to-peer ridesharing	
	Research and science	Citizen science · Open science (Access · Notebook science · Research · Science data · Plan S)		
	Data, information, and knowledge	Free content · Knowledge commons · Open communication · Open knowledge (Content · Data)		
	Communication and learning	Media	Collaborative writing · Democratic media · Open publishing · Participatory media · Peer review	
		Education	Open education (Educational resources · University (Admissions · Open-door academic policy))	
		Journalism	Citizen media (Citizen journalism and Wiki journalism) · Open-source journalism	
	Economy, production, and development	Products	Free and open-source software - FOSS (Free/Libre software · Open-source software) · Open-design movement (Robotics) · Open gaming · Open-source architecture · Open-source hardware	
		Economic principles	Gift economy · Open innovation · Open patent · Open standard · Sharing economy	
Politics and governance	Open government · Open-source governance			
Organizations	Creative Commons · Free Software Foundation · Open Architecture Network · Openmod initiative · Open Knowledge International · Open Rights Group · Open Source Initiative · Open Web Foundation · P2P Foundation · Pirate Party · PLOS · SPARC			
Activists	Tim Berners-Lee · Alexandra Elbakyan · Cody Wilson · Lawrence Lessig · Peter Murray-Rust · Douglas Rushkoff · Richard Stallman · Peter Suber · Peter Sunde · Aaron Swartz · John Wilbanks			
Projects and movements	DIYbio · Free-culture movement · Free software movement · Open science movement · Open Source Ecology · Open-source software movement · OpenCores · OpenWetWare · Sci-Hub			
	Tools	Licenses (Creative Commons · GPL) · Definition Docs (Free Cultural Works · Free Software · Open · Open Source) · Open Data Indices · Open educational resources		

Authority control  GND: 4548264-0 

Categories: [Computer law](#) | [Criticism of intellectual property](#) | [Free software](#)
| [Intellectual property law](#) | [Market structure](#) | [Open-source movement](#)
| [Social information processing](#) | [Standards](#)

This page was last edited on 4 March 2019, at 01:52 (UTC).

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Cookie statement](#) [Mobile view](#)





◀ **JOHN PERRY BARLOW LIBRARY**

A DECLARATION OF THE INDEPENDENCE OF CYBERSPACE

A NOT TERRIBLY BRIEF HISTORY OF THE ELECTRONIC FRONTIER FOUNDATION

A PLAIN TEXT ON CRYPTO POLICY

A PRETTY BAD PROBLEM

ACROSS THE ELECTRONIC FRONTIER

BARLOW IN ROCKSPACE

BARLOW, DENNING ON THE CLIPPER CHIP SCHEME

BEING IN NOTHINGNESS

COMPLETE ACM COLUMNS COLLECTION

CRIME AND PUZZLEMENT

CYNTHIA HORNER'S EULOGY

GO PLACIDLY AMIDST THE NOISE AND HASTE

IS THERE A THERE IN CYBERSPACE?

J. KREILSBURG-BARLOW INTERVIEW

JACK IN, YOUNG PIONEER!

JACKBOOTS ON THE INFOBAHN: CLIPPING THE WINGS OF FREEDOM

JUST SAY YES

LEAVING THE PHYSICAL WORLD

MITCH KAPOR & JOHN BARLOW INTERVIEW

PASSING THE BUCK ON PORN

SELLING WINE WITHOUT BOTTLES: THE ECONOMY OF MIND ON THE GLOBAL NET

SONGS FOR THE DEAD

STOPPING THE INFORMATION RAILROAD

TV, LSD, AND LIFE IN THE COUNTRY

THE ECONOMY OF IDEAS (WIRED MAGAZINE)

THE PURSUIT OF EMPTINESS: WHY AMERICANS HAVE NEVER BEEN A HAPPY BUNCH

THE VIEW FROM THE BROOKLYN BRIDGE

THINKING LOCALLY, ACTING GLOBALLY

THROUGH MANY PANES OF SHATTERED GLASS

TO BE AT LIBERTY

VERBUM MAGAZINE INTERVIEW

WHO HOLDS THE KEYS?

A Declaration of the Independence of Cyberspace

by John Perry Barlow

Governments of the Industrial World, you weary giants of flesh and steel, I come from Cyberspace, the new home of Mind. On behalf of the future, I ask you of the past to leave us alone. You are not welcome among us. You have no sovereignty where we gather.

We have no elected government, nor are we likely to have one, so I address you with no greater authority than that with which liberty itself always speaks. I declare the global social space we are building to be naturally independent of the tyrannies you seek to impose on us. You have no moral right to rule us nor do you possess any methods of enforcement we have true reason to fear.

Governments derive their just powers from the consent of the governed. You have neither solicited nor received ours. We did not invite you. You do not know us, nor do you know our world. Cyberspace does not lie within your borders. Do not think that you can build it, as though it were a public construction project. You cannot. It

is an act of nature and it grows itself through our collective actions.

You have not engaged in our great and gathering conversation, nor did you create the wealth of our marketplaces. You do not know our culture, our ethics, or the unwritten codes that already provide our society more order than could be obtained by any of your impositions.

You claim there are problems among us that you need to solve. You use this claim as an excuse to invade our precincts. Many of these problems don't exist. Where there are real conflicts, where there are wrongs, we will identify them and address them by our means. We are forming our own Social Contract. This governance will arise according to the conditions of our world, not yours. Our world is different.

Cyberspace consists of transactions, relationships, and thought itself, arrayed like a standing wave in the web of our communications. Ours is a world that is both everywhere and nowhere, but it is not where bodies live.

We are creating a world that all may enter without privilege or prejudice accorded by race, economic power, military force, or station of birth.

We are creating a world where anyone, anywhere may express his or her beliefs, no matter how singular, without fear of being coerced into silence or conformity.

Your legal concepts of property, expression, identity, movement, and context do not apply to us. They are all based on matter, and there is no matter here.

Our identities have no bodies, so, unlike you, we cannot obtain order by physical coercion. We believe that from ethics, enlightened self-interest, and the commonweal, our governance will emerge. Our identities may be distributed across many of your jurisdictions. The only law that all our constituent cultures would generally recognize is the Golden Rule. We hope we will be able to build our

particular solutions on that basis. But we cannot accept the solutions you are attempting to impose.

In the United States, you have today created a law, the Telecommunications Reform Act, which repudiates your own Constitution and insults the dreams of Jefferson, Washington, Mill, Madison, DeToqueville, and Brandeis. These dreams must now be born anew in us.

You are terrified of your own children, since they are natives in a world where you will always be immigrants. Because you fear them, you entrust your bureaucracies with the parental responsibilities you are too cowardly to confront yourselves. In our world, all the sentiments and expressions of humanity, from the debasing to the angelic, are parts of a seamless whole, the global conversation of bits. We cannot separate the air that chokes from the air upon which wings beat.

In China, Germany, France, Russia, Singapore, Italy and the United States, you are trying to ward off the virus of liberty by erecting guard posts at the frontiers of Cyberspace. These may keep out the contagion for a small time, but they will not work in a world that will soon be blanketed in bit-bearing media.

Your increasingly obsolete information industries would perpetuate themselves by proposing laws, in America and elsewhere, that claim to own speech itself throughout the world. These laws would declare ideas to be another industrial product, no more noble than pig iron. In our world, whatever the human mind may create can be reproduced and distributed infinitely at no cost. The global conveyance of thought no longer requires your factories to accomplish.

These increasingly hostile and colonial measures place us in the same position as those previous lovers of freedom and self-determination who had to reject the authorities of distant, uninformed powers. We must declare our virtual selves immune to your sovereignty, even as we continue to consent to your rule over our bodies. We will spread ourselves across the Planet so that no one can arrest our

thoughts.

We will create a civilization of the Mind in Cyberspace. May it be more humane and fair than the world your governments have made before.

Davos, Switzerland

February 8, 1996



WIKIPEDIA
The Free Encyclopedia

Not logged in [Talk](#) [Contributions](#) [Create account](#) [Log in](#)

[Article](#) [Talk](#) [Read](#) [Edit](#) [View history](#)



[Main page](#)
[Contents](#)
[Featured content](#)
[Current events](#)
[Random article](#)
[Donate to Wikipedia](#)
[Wikipedia store](#)

[Interaction](#)

[Help](#)
[About Wikipedia](#)
[Community portal](#)
[Recent changes](#)
[Contact page](#)

[Tools](#)

[What links here](#)
[Related changes](#)
[Upload file](#)
[Special pages](#)
[Permanent link](#)
[Page information](#)
[Wikidata item](#)
[Cite this page](#)

[Print/export](#)

[Create a book](#)
[Download as PDF](#)
[Printable version](#)

[In other projects](#)

[Wikisource](#)

[Languages](#)

[العربية](#)
[Català](#)
[Deutsch](#)
[Español](#)
[Français](#)
[עברית](#)
[Nederlands](#)
[Polski](#)
[Português](#)

A Declaration of the Independence of Cyberspace

From Wikipedia, the free encyclopedia

"**A Declaration of the Independence of Cyberspace**" is a widely distributed early paper on the applicability (or lack thereof) of [government](#) on the rapidly growing [Internet](#). Commissioned for the pioneering Internet project *24 Hours in Cyberspace*, it was written by [John Perry Barlow](#), a founder of the [Electronic Frontier Foundation](#), and published online on February 8, 1996, from [Davos](#), Switzerland.^[1] It was written primarily in response to the passing into law of the [Telecommunications Act of 1996](#) in the [United States](#). In 2014, the [Department of Records](#) recorded and released audio and video content of Barlow reading the Declaration.^{[2][3]}



Barlow at the 10th Anniversary of his "Declaration"

Contents [hide]

- [Content](#)
- [Background](#)
- [Critical response](#)
- [See also](#)
- [References](#)
- [External links](#)

Content [edit]

Governments of the Industrial World, you weary giants of flesh and steel, I come from Cyberspace, the new home of Mind. On behalf of the future, I ask you of the past to leave us alone. You are not welcome among us. You have no sovereignty where we gather.

— [John Perry Barlow](#), "A Declaration of the Independence of Cyberspace"

The declaration sets out, in sixteen short paragraphs, a rebuttal to government of the Internet by any outside force, specifically the United States. It states that the United States did not have the [consent of the governed](#) to apply laws to the Internet, and that the Internet was outside any country's borders. Instead, the Internet was developing its own [social contracts](#) to determine how to handle its problems, based

on the [golden rule](#). It does this in language evocative of the [United States Declaration of Independence](#) and obliquely cites it in its final paragraphs. Although the paper mentions the Telecommunications Act, it also accuses [China](#), [Germany](#), [France](#), [Russia](#), [Singapore](#), and [Italy](#) of stifling the Internet.^[4]

Background [edit]

At the time the paper was written, Barlow had already written extensively on the Internet and its social and legal phenomena,^[5] as well as being a founding member of the [Electronic Frontier Foundation](#).^[6] The work he was known best for previously, "[The Economy of Ideas](#)"^[7], published March 1994 in *Wired* magazine, also made allusions to [Thomas Jefferson](#) and some of the ideas he would write about in his declaration.

Critical response [edit]

Because of its subject matter, Barlow's work quickly became famous and widely distributed on the Internet. Within three months, an estimated 5,000 websites had copies of the declaration.^[7] At nine months, that number was estimated to be 40,000.^[8] To approach Barlow's vision of a self-governing Internet, a **virtual magistrate** was set up by the [Cyberspace Law Institute](#), now hosted by the [Chicago-Kent College of Law](#). [Magistrates](#) would be appointed by the institute and other legal groups to solve online disputes.^[7] The declaration has been criticized for internal inconsistencies.^[9] The declaration's assertion that 'cyberspace' is a place removed from the physical world has also been challenged by people who point to the fact that the internet is always linked to its underlying geography.^[10]

Outside the Internet, the response was less positive. [Larry Irving](#), the [Assistant Secretary of Commerce](#), said that a lack of safeguards would "slow down the growth of what is likely to be a major boon for consumers and business".^[7] In the online magazine *HotWired*, one columnist referred to his document as simply "hogwash".^[11]

By 2002 the number of sites copying the declaration was estimated to have dropped to 20,000.^[12] In 2004, Barlow reflected on his 1990s work, specifically regarding his optimism. His response was that "we all get older and smarter".^[13]

See also [edit]

- [Cyberspace](#)
- [John Perry Barlow](#)
- [Hacker Manifesto](#)
- [Commercialization of the Internet](#)
- [Surveillance capitalism](#)

References [edit]

- ↑ "John Perry Barlow: Is Cyberspace Still Anti-Sovereign?" ^[7]. 12 February 2018.
- ↑ "Department of Records" ^[7]. *www.departmentofrecords.co*.
- ↑ "Limited edition vinyl: John Perry Barlow reads "A Declaration of the Indenence of Cvherspace" / Boina Boina" ^[7]. *boinaboina.net*.

4. ^ Barlow, John Perry (1996-02-08). "A Declaration of the Independence of Cyberspace" [↗](#). Retrieved 2017-02-23.
5. ^ Ley, Michael. "DBLP: John Perry Barlow" [↗](#). Archived [↗](#) from the original on 25 February 2007. Retrieved 2007-02-09.
6. ^ "EFF: Board of Directors" [↗](#). Archived [↗](#) from the original on 6 February 2007. Retrieved 2007-02-09.
7. ^ ^a ^b ^c Yang, Catherine (1996-05-06). "Law Creeps Onto the Lawless Net" [↗](#). *BusinessWeek* (3474): 58–64. Retrieved 2007-02-08.
8. ^ Frezza, Bill (1996-11-01). "Can Public Network Computing Save Democracy". *Network Computing*: 35.
9. ^ Evans, Woody (2016). "Cyberspace is the Child of the Industrial Age - Defining it as Independent is Nonsense" [↗](#). *Institute for Ethics and Emerging Technologies*.
10. ^ Graham, Mark (2013-03-01). "Geography/internet: ethereal alternate dimensions of cyberspace or grounded augmented realities?" [↗](#). *The Geographical Journal*. **179** (2): 177–182. doi:10.1111/geoj.12009 [↗](#). ISSN 0016-7398 [↗](#).
11. ^ Cembalest, Robin (1996-09-20). "The Featherman File". *Forward*. **C**: 2.
12. ^ Barlow, John Perry (2002-01-22). "John Perry Barlow Declaration for Defendants in MGM et al. v. Grokster et al" [↗](#). Archived [↗](#) from the original on 10 February 2007. Retrieved 2007-02-09.
13. ^ Doherty, Brian (August–September 2004). "John Perry Barlow 2.0: The Thomas Jefferson of cyberspace reinvents his body — and his politics" [↗](#). *Reason*.

External links [edit]

- [A Declaration of the Independence of Cyberspace](#)
- [A Declaration of the Independence of Cyberspace \(Audio Recording\)](#)
- [A Declaration of the Independence of Cyberspace \(Portuguese Translation in PDF\)](#)
- [A Declaration of the Independence of Cyberspace \(Spanish Translation\)](#)
- [A Declaration of the Independence of Cyberspace \(official vinyl release with accompanying video and audio recordings read by John Perry Barlow and scored by Drazen Bosnjak\)](#)
- [Another Declaration Of Independence For Cyberspace which was conceived simultaneously and independently from the Barlow version. Note: archive.org time stamp is later than original publish date which happened during](#)



Wikisource has original text related to this article:
A Declaration of the Independence of Cyberspace



Wikiversity has learning resources about ***A Declaration of the Independence of Cyberspace***

HotWired/Weiner's Blue Ribbon Campaign where it hung and was blocked in the listserv for 1 hour when Barlow's version was delivered to the listserv using the same Subject line as this version from Citizens Of The Sea. [↗](#)

Categories: [1996 documents](#) | [Texts related to the history of the Internet](#)
| [Internet-related activism](#) | [Internet culture](#) | [Texts about the Internet](#)
| [Manifestos](#)

This page was last edited on 5 March 2019, at 05:28 (UTC).

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Cookie statement](#)

[Mobile view](#)



Ethereum Classic in 2019: An Interview with Anthony Lusardi



SFOX

Jan 24 · 23 min read



Following the enablement of Ethereum Classic on their algorithmic trading platform last month, the SFOX team was honored to sit down with Anthony Lusardi, the U.S Director of the Ethereum Classic Cooperative.

Lusardi has been involved with Ethereum Classic since its inception following the post-DAO-hack hard fork of Ethereum, advocating not only for ETC's continued development, but also for a healthy, positive relationship between it and ETH. In our conversation with Lusardi, we touched on everything from the value of immutability-at-all-costs, to the feasibility and utility of a truly decentralized crypto community, to ETC's recent 51% hack, to what 2019 might hold for ETC.

If you'd prefer to listen to this interview rather than reading it, check out the recording here:

SFOX: Thank you again for joining us, Anthony.

What ignited your passion for blockchain technology: what got you here working in blockchain tech and on Ethereum Classic?

Anthony Lusardi: First off, thanks for having me.

What got me interested in blockchain in general was nothing out of the ordinary. I think I first heard about Bitcoin when it crashed in 2013, and I said, “Well, that seemed like a good idea, but it looks like it’s dead now.” Then, in 2014 or so, I noticed that everything was still around; so, I did some stupid trading, lost some money, and then eventually learned about the tech and what the bigger, broader implications are for cryptocurrencies in general. And then I got into Bitcoin and I saw Ethereum, and Ethereum seemed pretty cool because you can do new things with it.

When the hard fork happened and Ethereum decided to reverse the DAO attack, I started supporting and volunteering on ETC. Then eventually, in February of last year, I took a full-time position at ETC Cooperative to just support the ETC ecosystem like I have been except maybe make some money—make ends meet, rather than just volunteering my free time.

SFOX: What led you to support Ethereum Classic during the DAO hack and subsequent hard fork, and why you believe that its principles of “decentralization and immutability above all else” are valuable things for a cryptocurrency to enact?

Anthony Lusardi: Those beliefs were shared by quite a lot of people. I would say actually at the time of the fork it was pretty evenly split 50/50 on whether or not people wanted to go with it, if you were just gauging sentiment and people’s comments online.

Obviously, the pro-fork side had quite a substantial amount of financial resources, and in the first week or two, I thought that all of Ethereum was going to have to go the way of the fork. And then I found out that somebody was actually running Ethereum Classic. So I joined the Ethereum Classic Slack, and I found that there were quite a lot of people there who believed the same things that I did about cryptocurrencies in general: that cryptocurrencies need to function as they’re programmed to, according to the rules that were agreed to—and that those rules don’t change, like Ethereum wanted through the fork.

When you change the rules, you end up in a situation where the members of your bank can vote on your bank balance. Would you really want everybody within your particular bank to be able to say, “Oh, Aaron did something bad, and we just looked at it for five days and all agree that Aaron did something bad, so we’re gonna just take all of Aaron’s money away?” Or maybe you did do something bad, maybe you

take all of Aaron's money away"? Maybe you did do something bad, maybe you didn't, but we don't believe that cryptocurrencies should function in that particular way; that's why I, personally, have been supporting ETC.

You'll see there are quite a lot of very good reasons to support things like immutability—things like censorship resistance, in particular—throughout the space, even in the largest cryptocurrency there is: in Bitcoin, you see that very, very strongly. Other cryptocurrencies are going to exist, and people are going to have different needs and wants for them, so it's very important that we keep those principles in our other cryptocurrencies, from my point of view.

SFOX: Let's dive into this idea of immutability a little further. One of the things that I struggle to understand as I think about these concepts and crypto is just how far they go and the extent to which we ought to adhere to them. On the one hand, "immutability" sounds like things should never be changed; on the other hand, as you just mentioned, the decision regarding the DAO fork was made in about five days. That leads me to wonder whether the idea behind Ethereum Classic is (1) things should never be changed or (2) things should just be changed very slowly and conservatively.

It's interesting context for us at SFOX to be thinking about this because, at the same time that we enabled Ethereum Classic on SFOX, we also enabled Bitcoin SV, which as you probably know, is all about bringing Bitcoin back to what that camp believes is the original vision of it, as outlined in Satoshi's white paper and his other early writings. And so that leads me to wonder: Is Ethereum Classic a kind of "Vitalik Vision," in terms of just enacting the "original vision" of Ethereum and keeping it that way, or does it rather say, "Yes, we're going to make changes over time, but they're going to be very conservative and slow-moving changes"?

Anthony Lusardi: Yeah, so there's a couple of things there.

One: You'll see in the Bitcoin community, the ETC community, and other communities that there isn't actually a leader or a centralized source of ideas, and that's a really important property that we're trying to maintain. When you look at things like Bitcoin Satoshi Vision, for example, they are very much appealing to authority: They're very much wanting to go with whatever their readings of the gospel of Satoshi are. That's not actually something that we share—there's quite, quite a significant amount of differences there. But it is interesting

quite a significant amount of differences there. But it is interesting.

You mentioned immutability and what that means in terms of upgrades: whether it should be slow or fast, and what it should be. First and foremost: When we say “immutability”—or, okay, when *I* say “immutability,” because I can’t speak for everybody—I mean *immutability of state*. Proof of work is a kind of probabilistic guarantee that state isn’t going to be changed, and the longer a particular block is on the network, the more likely it is to never be changed. I think that that’s the type of property that ETC, and ETH, and every other cryptocurrency out there should actually have. And it’s kind of interesting because that property, in particular, was violated when the hard fork happened: they didn’t go and start mining at a pre-DAO block in order to advance the chain and get ahead of it to the point where it didn’t exist on the chain anymore. That just kind of superseded all of the assumptions of proof of work and undid something in an unacceptable way.

When you say “immutability,” sometimes people think that that means the blockchain is never going to upgrade, or it’s going to get stale, or it’s going to get old. It doesn’t mean any of that. It simply means that if Aaron’s account balance says that Aaron owns 10 ETC, then nobody is allowed to change that on the blockchain except for Aaron, who owns his private keys. And in that way, *you’re* the only one who’s able to change the blockchain to change your account balance. When you allow other people to vote on it, that’s the thing that we take issue with; we think that immutability needs to be preserved there.

SFOX: I noticed, Anthony, that you were very careful to say that that’s immutability as *you* see it; similarly, in your writing about issues like decentralization and immutability, you emphasize that the communities of blockchains such as Ethereum Classic ought to be decentralized and that, therefore, you can’t speak for anyone other than yourself.

I wonder whether a blockchain with truly decentralized leadership—as, it seems, you envision ETC—can really develop with the kind of cohesion that a blockchain, or really any kind of project with more centralized leadership, can. I found even in my own research trying to learn about ETC that it was much harder to dive into it and understand the collective ecosystem than it is to learn about a more centralized blockchain, because there are just so many different projects and so many people in the mix when it comes to ETC. Could speak to that notion you have of decentralized leadership and decentralized communities. and why you believe that it’s a good and productive thing for

communities, and why you believe that it's a good and productive thing for blockchains like ETC?

Anthony Lusardi: I think that's a really good point.

Obviously, any small, centralized project is going to be able to move faster and implement whatever particular vision or goal they want—but there are a lot of weaknesses that are inherent there. There's a lot of unfairness, a lack of equity, between members participating in it, because when you run a centralized program, you end up beholden to whatever the developers decide to change or do with it.

Obviously, it's a double-edged sword. There are some benefits to centralization in that you can clearly present the “one true kind of direction,” or you can take a position very easily on what the proper path forward is. With something like ETC or Bitcoin, you don't get that convenience, but what you do get is a substantial amount of strength in the lack of dictators and the lack of having a central, core party that can control every part of it. And I think that's really interesting and important.

I'm not talking about any blockchain in particular, but I do think that, in general, the more decentralization and distribution you can get in your community, the more you can give people recourse in the event of a bad action—even just a “disagreeable” action—taken by one party, the better off it all is.

SFOX: Let me push on that idea a little bit.

Looking around in the crypto world, one of the things that is really interesting and challenging to me is exactly what you alluded to: the de facto “dictators” of the community, especially when we live in an age where so much crypto discourse is borne out through media like Reddit or Twitter, where there are giant figures who wield so much influence. But then I wonder about other models of governance—not even within crypto or blockchain, but something like the U.S.'s democratic government. Imagine some sort of government structure where it *does* seem that there are defined and somewhat centralized authorities, whether that's the President, or the House, or the Judicial Branch. When it's working well, the notion of democracy is that there is this firm structure in place that represents the interests of everyone in a reasonably fair way, right?

With that model in mind, do you imagine that there could eventually be that kind of “middle ground” that exists in a blockchain community where there is

kind of “middle ground,” let’s call it, in a blockchain community, where there is a kind of defined and centralized government that does a good job of representing everyone’s interests—or, do you really believe that in order for blockchains to fulfill the underlying vision of the technology, their governmental structures and communities have to be truly decentralized?

Anthony Lusardi: Well, I guess it depends.

Funny enough, I was in Singapore in September, and even though I found it to be more of a Vegas-y atmosphere, I thought it was really nice that everything was so well-kept and neat. And from a general standpoint, beyond just the aesthetics, you could say, “Singapore is great.”

Singapore has a benevolent dictator, and benevolent dictators sometimes work out really, really well—you know, for short periods of time. And then the benevolent dictator gets corrupted, or the benevolent dictator’s son or grandson gets corrupted; all of the sudden, things change, and having a dictator isn’t so great anymore.

So in general, I think that, yes, centralization does make certain things easier—there’s no arguing that. It makes both good *and bad* things easier, though. And what I would say is, in general, when you look at projects throughout the cryptocurrency space, you will find that the overwhelming majority of them are centralized: the overwhelming majority of them have a very small group that can just steer the entire direction without much recourse to anybody else. And if you’re looking for the opposite of that—if you’re looking for a cryptocurrency that says, “No, this isn’t the way that it should be governed, this isn’t the way that this should happen”—they are few and far between.

I think that the market space or the market opportunity—or maybe not market, but the mindshare opportunity—for decentralized cryptocurrencies, for ones where there is no one true owner, is actually very substantial. There’s a lot of potential there, and that’s really what we want to try to do: We want to push to get people to better understand that centralization isn’t the only option. And in something like a cryptocurrency—where, if we’re ever successful, we will have an immense amount of people using it—it’s vitally important that we avoid the type of centralized structure where one government can have an undue influence over the monetary policy.

We already see that “undue influence” with USD, for example: If the U.S. doesn’t like

you they don't even need to pass laws against you, they don't need to go to war with you—they just need to cut you off from their banking.

SFOX: With that in mind, let's pivot more specifically to the decentralized community of ETC. Like I mentioned, as I was trying to learn about ETC, I encountered the difficulty of holding all of the different pieces of its community in my mind and seeing how they fit together. What are the major organizations involved in ETC right now, and how are they interrelated?

Anthony Lusardi: Sure. So, in general, I would call it a *distributed community*. It can't really be decentralized; we just try to maximally distribute it so nobody has control over all the communications channels.

There's us at the [ETC Cooperative](#), and we do a lot of ecosystem support for ETC. We write a lot of guides and try to talk to developers and raise interest in ETC in general, in all the ways we can. We provide certain third-party services—such as block explorers, or RPC nodes, or even just regular nodes—so that people have an easier time using ETC when they need to.

There is [ETC Labs Core](#): the majority of their employees are former [ETCDEV](#) employees—unfortunately, that organization closed—and they do a lot of the core development for the network. So, the coming opcode upgrades are going to be largely implemented and done by that team. They're also working on a lot of really neat tools. They just published [an article](#) yesterday where they're going to speed up the EVM; they're working on some IDE stuff; they're working on something called LLVM... We don't need to really go into the technicals there, but there's a lot of interesting stuff coming out of them.

Then we have [IOHK](#), and they do a lot of research for ETC. They—again, this is highly technical and even way beyond me, so everybody reading is in the same ballpark here—but they've formalized the EVM in K, and they've done a lot of other very interesting research. They wrote the Mantis client, too, which they'll be upgrading for the various opcode upgrades that are going to happen on ETC probably sometime in the next 3 or so months.

There are other teams that have come and gone, and there are tons of independent developers that do various different things. It's a really big ecosystem. And if you wanted to learn more about some of the different projects and different teams

working on ETC, then I would just encourage you to start at [etherbase.io](#) and then

working on ETC, then I would just say you can start at etnereumclassic.org and then just go from there. Check out [our subreddit](#); check out [our discord](#); check out [our telegram group](#). There's a lot of stuff there.

SFOX: You mentioned that there are teams that have come and gone—in particular, as you said, ETCDEV unfortunately had to shut down on December 3rd. And then just yesterday, ETC Labs announced that they're launching this new initiative, ETC Labs Core. Is it just part of the distributed nature of ETC that we should expect these kinds of institutional turnovers over time? Or, do you see that as a symptom of something else, and, in the long run, will there be a number of distributed but established and longstanding ETC organizations?

Anthony Lusardi: So actually, at ETC Cooperative, we are talking to other groups and trying to get them to do some coding and some development for ETC. Obviously, we want to have as many groups working on as many things for ETC as possible. And, yeah, the turnover, it's natural, and it happens sometimes. You know, like I said, ETCDEV folded, but all their devs were skilled in developing stuff for ETC, and they largely went to ETC Labs Core.

Turnover isn't great: it sucks if people lose their jobs, and it sucks if there are, you know, issues that come from that. But the *good thing* is that there is no centralized team that can go and say, "I'm going to use ETC to bail myself out," and you do see that happen in other cryptocurrencies, where the core team ends up in a situation where they were either financially irresponsible, or they had some other issue, or they became highly centralized and lost a lot of their support—and they can't just turn around and cash out on the equity in their blockchain. They have to, you know, survive like any other business when in the natural world. If you have a startup and that fails, you can't go to the bank and try to change your account balance or do other fishy stuff in order to bail yourself out.

And we do see those sorts of bailouts in other blockchains. For example, there was one—I'm not going to name them, but they essentially tried to develop their own cryptocurrency miners for their network and found that there was a lot of competition around this: two other groups had developed cryptocurrency miners for their network as well. Because the core team had made one of the miners, even though they weren't able to keep up—even though they weren't able to complete their orders and were unable to meet the deadlines that they had set for themselves—they just decided that they were going to make it so that only their own cryptocurrency miners could

decided that they were going to make it so that only *their* cryptocurrency miner works on their network, effectively removing the rest and working in a way that gives them a very unfair advantage and is very bad for the network in general, because those unfair advantages tend to lead towards centralization.

SFOX: So it sounds like, to pivot back to what we were talking about earlier, one of the virtues of the distributed community of ETC is that it helps to support the “free market” idea of only allowing the successful businesses to stay around in the long run. Is that right?

Anthony Lusardi: Yeah. And you know, that’s just a fact of life. And it’s somewhat about free market, and, you know, the successful businesses being able to support themselves, but it’s more about the fact that there is an *even playing field* for everybody.

It’s a true, free market in that anybody who wants to participate in ETC gets the same exact playing field as everybody else, and there’s nobody there who’s going to be able to do things for themselves that other people will not be able to. There’s no Chancellor on the brink of bank bailouts in 2008, where banks, even though they did some bad things and should have failed, were not even allowed to fail by the financial system. And if anybody missed that reference, it goes back to the first Bitcoin block, which actually has that message encoded from Satoshi because that was, probably, the most original purpose of cryptocurrency: that everybody gets an even and fair playing field.

SFOX: When the ETC Cooperative talks about how it was founded, it says that Grayscale agreed to contribute a third of the Ethereum Classic Investment Trust’s management fee to the Coop for three years from April 2017 onward. And so that makes me wonder, as we’re talking about the transience and turnover of organizations: Is the ETC Cooperative just intended to be a temporary organization for three years or so? And if that’s not the case, how does it plan to fund its operations after that initial appropriation ends?

Anthony Lusardi: No, we intend to operate for much longer than that. The three years is kind of some somewhat-guaranteed starting money that can really help to bootstrap the ETC Cooperative, but we are also now a 501(c)(3) non-profit organization. So we’re going to be essentially looking for donations and other forms of funding to keep us going, and we have no intention of stopping any time soon.

SFOX: That is an interesting model in this cryptocurrency world that seems to be, in so many cases, for-profit: you're actually looking for donations, on the other side of it.

Anthony Lusardi: Yeah, and without selling a token. ...That was a mean dig.

SFOX: *Laughter* Fair enough.

In terms of fiduciary responsibility: is the ETC Cooperative bound by any actual legal responsibilities to make prudent investments, or is that more of just a principle that you're trying to uphold?

Anthony Lusardi: We're only supporting the ETC ecosystem. We don't give financial advice to anybody; I even personally don't give out financial advice in general. We're not we're not selling ETC, we don't hold any substantial amounts of ETC, and we're not trying to encourage people to do anything with ETC financially: whatever they do is up to them. We just want to support the ecosystem, help it develop and mature, and keep these core principles of the ecosystem alive.

SFOX: Moving the matter of ETC vs. Ethereum: these two chains have had an interesting history and relationship since they forked, and I know that in recent times you've talked about how the ETC community has put a lot of work into finding ways for those communities to work together. As it stands right now and looking ahead, how would you say that ETC and Ethereum differ in their approaches to security, immutability, and the other core principles that define a blockchain?

Anthony Lusardi: The major way we differ right now is that Ethereum is going to proof of stake and ETC is going to go proof of work, and those are two completely different kinds of consensus mechanisms and security mechanisms. And I would say that's the major way we disagree.

But also, to your point, I consider quite a good deal of people in "the Ethereum community" to be friends at this point. So, yeah, there's definitely been a lot of healing between the two communities since the fork. We disagree quite a lot on ideology in some cases, but I think one of the interesting things is that when somebody is my friend, I have a much easier time convincing them of my point of view than I do if they're my enemy—so I'm looking to hopefully see that all continue.

SFOX: Sure. And when Twitter, bad conversations, and negative energy are so prevalent in the crypto community, I think that positive attitude that you've just evinced is so important in moving the overall industry forward. So, respect to you for that.

Anthony Lusardi: Thank you. I don't deserve it all the time, though—there are definitely times where I'm not positive on Twitter!

SFOX: Well with so much negativity out there, sometimes you just *have* to give in, right? But—understood.

Something that's also interesting to me is the notion of *metrics* in crypto and how we all can be responsible in both measuring and reporting those metrics. The ETC Cooperative says that it's going to monitor and periodically evaluate its success in meeting its different objectives. So I'm wondering how the Coop knows that it's going to be successful or that it has been successful, when so many metrics are dangerously gameable—metrics like market cap or the social communities that can be, as you know, bought or sold to create a perception of success or failure that doesn't align with reality.

How do you deal with metrics, and which ones do you think are the most important or reliable for a cryptocurrency like ETC and a coop like the ETC Coop?

Anthony Lusardi: In general for the ETC Cooperative, we don't look at metrics like market cap, or really anything that's market-based, for some measure of success. What we do is we look at, you know, how many developers we reached. We look at how many projects we helped out. We look at whatever our major accomplishments were for the previous six months or so, and that's how we measure how well we're doing, where we can improve, and where we can do better. So, for example, if you go on etccooperative.org, you'll find [our review published mid-year last year](#), and we need to publish another end-of-year review very soon—and that would give you further insight on how we measure our success.

How does ETC measure success? I think it's going to vary by person, by group, and by project. Obviously, there are lots of people who are going to measure success in dollar terms. There are people who are going to measure success in total number of users. There are people who are going to measure success in adoption and usage by companies. You know, there are a lot of different ways to measure success. But

by companies. You know, there are a lot of different ways to measure success. But yeah, in general, I just don't like doing it by market cap or by other easily gameable metrics like that.

SFOX: A while back, some people said that different coins are monopolies and so, eventually, everyone would move to Ethereum and ETC would die. We've seen, clearly, that that is not the case, because ETC is still here and growing. What do you think has helped ETC to continue to grow, and how do you think ETC will find the way forward to continue to grow and compete from here on out?

Anthony Lusardi: You know, ETC has a very strong contingent of passionate supporters, and they've definitely helped us grow. In general, I think just sticking to your guns and delivering on what you said you were going to deliver on is vitally important and helps you grow and succeed in this industry.

As far as cryptocurrencies surviving and dying off, I mean, I don't know. I think there's quite a great deal of cryptocurrencies that will die off, even if they're good ideas, as a consequence of the market kind of condensing on things. But, yeah, in general, I think that ETC has been successful because it sticks to its guns, does exactly what it says on the tin, and doesn't deviate from that path.

SFOX: Before we wrap up, I have to ask you about the recent attack that ETC had on its network. What lessons do you think the community can take from it? How do you think the community—with the caveat that you can only really speak for yourself—intends to approach the risk of future such attacks?

Anthony Lusardi: Yeah, I mean, for myself, we intend to approach it by essentially following the same rules that ETC always has. We are going to definitely step up monitoring of the ETC blockchain—maybe not ETC Cooperative, but many people in the community—and then on top of that, we're looking at ways to defensively mine or ways to mitigate 51% attacks.

One of the very interesting things about a 51% attack is that when you look at exchanges in general, the amount of confirmations—the amounts of blocks they wait before considering a transaction final—varies quite a lot. So on Bitcoin, for example, you'll wait two confirmations, and, if you had the hash power, you could undo two Bitcoin transactions with about \$80,000 USD worth of bitcoin. Or you could spend, you know, another \$120,000 USD and kind of guarantee that your blockchain gets

ahead and removes the old transactions, allowing you to double-spend an exchange. \$120,000 USD isn't that much when the payout is substantially higher—in the tens to even hundreds of millions.

But when you look at ETC, for example, you know, people wait about 75 or so confirmations. That amounts to \$1,500 USD to \$2,000 USD worth of mining, which is a substantial decrease in the total security of your deposits. So what we've really been encouraging people to do is to actually increase confirmations for ETC: make sure that deposits take longer, because the longer the deposit takes, the more it costs to undo it. And if it costs more to undo the deposit than the attacker is going to get back, then it's just not worth it.

We really just want to see exchanges at least use the same amount of security for ETC that BTC has, for example. So, for ETC, we've been suggesting that people wait 2,500 to 5,000 or so confirmations, especially depending on the value of the transaction. You know, if it's a low-value transaction then you can wait less; if it's a high-value transaction, then you can wait more. And I think educating people on that particular property of proof of work is very important, too, because I think a lot of people don't realize that confirmations can sometimes be undone rather trivially if you don't get enough of them.

SFOX: So it sounds like the future security of ETC—and other cryptocurrencies, I imagine—depends on the entire ecosystem working together: miners, but also exchanges, in terms of how long they decide to wait, how many confirmations, and things like that. Is that right?

Anthony Lusardi: Yeah, exactly. And that's not to say that Bitcoin or any other proof-of-work cryptocurrency is "weak." It's just that the risks and security of these cryptocurrency networks need to be well understood and I think, right now, there's a lot of people who are actually secretly—or, in some cases, outwardly—very happy that various proof-of-work networks are under these 51% attacks.

I want to just communicate to anybody reading this interview that proof of work is the only way you get a decentralized cryptocurrency, and if you want that, then you need to properly understand the risk of them. And I think that right now, there's a lot of people who are kind of gleefully ignoring a potential risk further down the line for the short-term bad press that some other cryptocurrencies can get if they're 51% attacked. Really, the 51% attack is never on the network: The network will settle out

and will probabilistically advance towards an immutable consensus.

But this is really just hard to explain in simple language—it's a really complicated issue.

SFOX: Yeah. But the interesting point in there, I think—or one of them—is a sentiment about which Charlie Lee tweeted a week ago: “Look, if a cryptocurrency is not susceptible to a 51-percent attack, then it's not decentralized.” It sounds like you agree with that statement, right?

Anthony Lusardi: Yeah, exactly. So, yes, ETC, for example, got 51% attacked, but centralized cryptocurrencies can get *1% attacked*: they could get attacked by an even smaller group of people with even less money. So, decentralized is always better.

SFOX: What do you think ETC's biggest hurdle in 2019 will be? And, on the other hand, what are you most excited about for ETC's continued development in the coming year?

Anthony Lusardi: I honestly think the biggest hurdle is properly informing exchanges and other people transacting on proof-of-work cryptocurrencies as to what the threat model is, so that people understand how to properly use them.

The things I'm looking forward to, really, are a lot of tools and a lot of development around ETC that should make development a lot easier—should be much more attractive, should maybe get the EVM executing a lot faster so that smart contracts are more efficient and the node-syncing is better. Yeah, there's quite a lot of technical-level advancement that I'm really hoping to see an ETC this year.

Thanks to Anthony Lusardi for offering us his perspective on crypto, Ethereum Classic, and the latest in blockchain innovation.

Get the SFOX edge on your largest ETC orders by trading algorithmically across multiple exchanges directly from your SFOX account.

Got another minute? Check out:

What is Ethereum Classic?

On July 20th, 2016, the future of Ethereum changed, basic assumptions about blockchain were challenged, and a new cryptoasset was born.

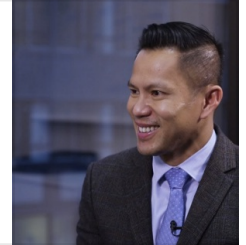
blog.sfox.com



Bitcoin SV in 2019: An Interview with Jimmy Nguyen

SFOX sat down with Jimmy Nguyen to discuss Bitcoin SV, hard forks, and the path forward in crypto.

blog.sfox.com



The above references an opinion and is for informational purposes only. It is not intended as and does not constitute investment advice, and is not an offer to buy or sell or a solicitation of an offer to buy or sell any cryptocurrency, security, product, service or investment. Seek a duly licensed professional for investment advice. The information provided here or in any communication containing a link to this site is not intended for distribution to, or use by, any person or entity in any jurisdiction or country where such distribution or use would be contrary to law or regulation or which would subject SFOX, Inc. or its affiliates to any registration requirement within such jurisdiction or country. Neither the information, nor any opinion contained in this site constitutes a solicitation or offer by SFOX, Inc. or its affiliates to buy or sell any cryptocurrency, securities, futures, options or other financial instruments or provide

cryptocurrencies, securities, futures, options or other financial instruments or provide any investment advice or service.

Interview

Crypto

Ethereum

Blockchain
Technology

Bitcoin



397 claps



SFOX

Prime dealer of cryptoassets for sophisticated traders and institutional investors.

Follow



SFOX Edge

Institution-Grade Crypto Insights and
Analysis

Follow



Never miss a story from **SFOX**
Edge

GET UPDATES



WIKIPEDIA
The Free Encyclopedia

Not logged in [Talk](#) [Contributions](#) [Create account](#) [Log in](#)

[Article](#) [Talk](#) [Read](#) [Edit](#) [View history](#)

[Main page](#)
[Contents](#)
[Featured content](#)
[Current events](#)
[Random article](#)
[Donate to Wikipedia](#)
[Wikipedia store](#)

[Interaction](#)

[Help](#)
[About Wikipedia](#)
[Community portal](#)
[Recent changes](#)
[Contact page](#)

[Tools](#)

[What links here](#)
[Related changes](#)
[Upload file](#)
[Special pages](#)
[Permanent link](#)
[Page information](#)
[Wikidata item](#)
[Cite this page](#)

[Print/export](#)

[Create a book](#)
[Download as PDF](#)
[Printable version](#)

[Languages](#)

[தமிழ்](#) [Edit links](#)

Rare earth industry in China

From Wikipedia, the free encyclopedia

The **rare earth industry in China** is a large industry that is important to Chinese internal economics. **Rare earths** are a group of elements on the periodic table with similar properties. Rare earth metals are used to manufacture everything from electric or hybrid vehicles, [wind turbines](#), consumer electronics and other clean energy technologies.^[1] The elements are also important to national governments because they are used in the defense industry.^[2] Twenty percent of rare earth demands are for use as permanent magnets.^[2] Permanent magnets can be used for a variety of applications including serving as essential components of weapons systems and high performance aircraft.^[2]

Rare earths are found in various minerals such as monazite and bastnasite.^[3] They are dispersed in low concentrations and are costly to extract from ore.^[4] Major reserves in the world exist in China, California, India, Brazil, Australia, South Africa, and Malaysia.^[3] However, China accounts for over 95 percent of the world's production of rare earths.^[5] Therefore, having control of these elements puts China at a powerful position.

It is estimated the world has 99 million tonnes of rare earth reserve deposits.^[6] China's reserves are estimated to be 36 million tonnes or roughly 30 percent of the world's total reserves.^[6]

Contents [\[hide\]](#)

- [History](#)
- [Research](#)
- [Controversy and Political Implications](#)
- [Major players](#)
- [Environmental Impact](#)
- [See also](#)
- [References](#)
- [External links](#)

History [\[edit\]](#)

In 1927, rare earths were discovered and small scale production of concentrates started as early as 1958, but the government did not have interest in the large scale potential of these elements until the 1980s and 90s.^[5]

In China, [Xu Guangxian](#) is regarded as the founding father of China's rare earth industry. A pioneer for rare earth research, Xu Guangxian came back to Peking University in China after graduating from Columbia University.^[3] In 1980, he joined as a member of the Chinese Academy of Sciences.^[7] A few years later, Xu created

the State Key Laboratory of Rare Earth Materials Chemistry and Applications for research on these elements.^[3] Xu would go on to have powerful positions in the scientific community such as a Director in the National Natural Science Foundation of China, Chairman of the Chinese Chemical Society, and Vice Chairman in the Chinese Society of Rare Earths.^[7] Later in the 2000s, Xu was also influential in telling the government to adopt export quotas because he saw the potential rare earths had in the technology sector and wanted to keep these precious resources within China.^[8]

Additionally, in 1980, the Chinese Society of Rare Earths was created and just five years later, they established the China Rare Earth Information Center (CREIC).^[3]

Government support increased in 1986 with the program known as [Program 863](#) which calls for the advancing of the country through technological breakthrough and increase research to propel the country forward economically and strategically.^[9] Another important program [Program 973](#) was created as a basic research program to utilise science to advance the rare earth industry.^[9] During this time, the government poured funding into the resources and knowledge that were essential to the establishment of the industry.

Another form of governmental support was its encouragement to Chinese rare earth companies to partner with rare earth industries from other countries.^[3] In 1979, Japan's Inoue Japax Research worked with China to do research regarding ore analysis, ore dressing, and product application.^[3] In 1989, Ke Ning Da Industry of Ningbo, China partnered with US firm Tredas International to make 40 tons of magnets.^[3] The Chinese rare earth network later expanded to include relationships with Canada and additional countries from the West.^[3] During these ventures, the Chinese government provided more money for new facilities and the industry also received new technologies from their partners which catapulted China to the forefront of rare earth production.

In 2002, China's central government pushed forward restructuring of the domestic rare earth industry by creating two state-owned groups China Northern Rare Earth Group Company and China Southern Rare Earth Group Company.^[10] This largely failed due to opposition from powerful local authorities and local producers.^[10] Fierce competition in the local sector produced low profitability and inefficiency. This drove producers to consolidate and merge into larger companies for survival.^[10] Market forces thus accomplished what [central planning](#) could not.

As rare earth prices went up because of the restriction of export, many [illegal mines](#) were developed by organized criminals to benefit from the trade.^[11] The smuggling by organized criminal groups is harmful to China's rare earth industry as it depletes resources rapidly, deflates prices and causes supply problems for local producers.^[12] It is estimated a third of exports or 20 000 tonnes in 2008 were illegally exported from China.^[12]

It is said China contains 36 percent of the rare earth deposits in the world.^[13]

Due to Chinese export restrictions and heavy dependence of foreign countries on Chinese sources, efforts are ongoing to restart rare earth industries in other countries and to pressure countries with intensive industry, like Japan, to source rare earths elsewhere.^[14] Non-Chinese companies which will benefit from increased

demand are Lynas Corporation and Alkane Resources of Australia.^[14] The [Mountain Pass mine](#) in California which has been closed since 2002 due to heavy competition from China will be restarted by Molycorp.^[15]

It has been reported that Chinese authorities will set up an industry group called **The China Rare Earth Industry Association** to coordinate pricing collectively with foreign buyers.^{[16][17]} Wang Caifeng will be the chief of this industry group, which is expected to be formally established in May 2011.^{[16][18]}

Research [\[edit\]](#)

China has two state research facilities which provide specialized research into rare earth elements.^[19] They are the Rare Earth Materials Chemistry and Applications state key laboratory, which is associated with Peking University,^[19] and the Rare Earth Resource Utilization state key laboratory located in Changchun, [Jilin](#) province.^[19]

The Chinese rare earth industry also has two journals which publish research in rare earth minerals.^[19] They are the Journal of Rare Earth and China Rare Earth Information (CREI) Journal.^[19] These journals are published by the Chinese Society of Rare Earths established in 1980 by Chinese rare earth researchers.^[19]

Controversy and Political Implications [\[edit\]](#)

From 2000 to 2009, China's production of rare earth elements increased 77 percent to 129,000 tons while production from other reserves dropped to around 3000 tons.^[5] Large US mining companies such as Molycorp closed due to the mix of China's abundance of rare earths and their capacities for production, the cost of labor, and stringent environmental regulations during the Nixon era.^[3] With the decreased pool of competitors, China's hold on these elements gave them a lot of power in the distribution of these commodities. The government declared these elements to be a protected and strategic good in 1990.^[5] This decision had a significant impact on foreign industries who partnered with China. Foreign investors could no longer work with rare earths except when partnered with Chinese firms.^[5] The State Development and Planning Commission gained power as all projects needed their approval.^[5] Production quotas were instigated for the miners and oftentimes quotas would be surpassed because of illegal mining by people who did not have licenses.^[5]

The Chinese government was also able to use these rare earths as a means of exerting power over other countries. As production levels reached all time highs and China declared rare earths to be protected, the government imposed tight regulations on exports. Currently, the Ministry of Commerce is responsible for setting quotas for domestic producers and for foreign joint venture producers.^[5] In 2015, a select 20 domestic producers could export elements and the export quota was 35,000 tons total for both domestic and joint venture producers.^[5] These decreasing figures alarmed other countries because they depend on China for their supply of rare earths. If China were to cut off the exports, the results for the technology sector would be disastrous. This occurred temporarily in 2010 when the Chinese had tensions with Japan due to a maritime dispute.^[3] They stopped all their

exports to Japan and also reduced their exports from 40 to 30 percent.^[3] China demonstrated to the world that they would use this tactic as a means of coercion should need be. In response, the US and Japan appealed to the World Trade Organization to reduce their practices that secured the monopoly on rare earths and to stop pressuring other countries to move their jobs to China.^[3] The appeals did not induce a significant change in the way the Chinese government controls these elements today.

China's rare earth industry is of significance to the United States and the rest of the world today because of the increasing demands for tech products. Tesla's shift to a magnetic motor for its Model 3 Long Range car will catapult sales for neodymium.^[20] Because of the export quota, prices for rare earths are rising. The current cost for 1 kilogram of neodymium is 70 US dollars.^[20] However, it is expected to increase as 3,300 of the 31,700 tons of global demands were not met in 2017 and it is calculated that the demand will increase to around 39,000 tons by 2019.^[20] Therefore, countries are going to have to find ways to reduce rare earth usage, mine their own, or pay ever increasing prices.

In addition, political relations play a large factor in the distribution of these commodities. In 2018, US President Donald Trump proposed tariffs on technology products imported from China.^[21] As a result, China immediately responded with tariffs on US goods.^[21] Were the Chinese to impose sanctions on these elements just like what happened with Japan in 2010, the US technology sector would be greatly harmed. US companies such as Apple and Boeing get the materials and labor from China and a significant part of their profits come from China's large market.^[21]

Major players [edit]

The Chinese rare earth industry is dominated by local state owned firms, private firms and centrally owned state firms.^[22]

In northern China, rare earth industry is dominated by the Inner Mongolia [Baotou Steel Rare-Earth Hi-Tech Company](#). In southern China, [China Minmetals Corporation](#) is the dominant player in the region.^[22] Other major players include the [Aluminum Corporation of China Limited](#) and [China Non-Ferrous Metal Mining](#).^[22]

Environmental Impact [edit]

The US stopped their mining operations in part because of the immense pollution from the practices to extract these rare earths. However, as the main producer for the world, China did not waver but rather increased their production levels. The major cities in which rare earths were mined are Shandong, Inner Mongolia, Sichuan, Jiangxi, Guangdong, Fujian, Hunan, and Guangxi.^[2] This has caused lasting damage to the villages surrounding the factories.

The sewage produced from the factories were dumped into nearby ponds and rivers.^[23] According to accounts from a resident of Bayan Obo, a major production center, "Before the factories were built, there were just fields here as far as the eye can see. In the place of this radioactive sludge, there were watermelons, aubergines

- Chinese move" [↗](#). *The Australian*.
15. ^ Suzanne Goldenberg (1 January 2011). "US digs deep to secure the technology of the future" [↗](#). *smh.com.au*.
 16. ^ [a](#) [b](#) "China setting up rare earth industry group" [↗](#). *chinapost.com.tw*. 29 December 2010.
 17. ^ Luo (28 December 2010). "China to Set up Rare Earth Industry Group" [↗](#). *CRIENGLISH.com*.
 18. ^ "China Setting Up Rare Earth Industry Group" [↗](#). *industryweek.com*. 28 December 2010.
 19. ^ [a](#) [b](#) [c](#) [d](#) [e](#) [f](#) Cindy A. Hurst. "China's Ace in the Hole: Rare Earth Elements" [↗](#). National Defense University Press.
 20. ^ [a](#) [b](#) [c](#) Editorial, Reuters. "Tesla's electric motor shift to spur demand for rare earth neodymium" [↗](#). U.S. Retrieved 2018-04-04.
 21. ^ [a](#) [b](#) [c](#) "Trade war with China could hurt these U.S. businesses most" [↗](#). Retrieved 2018-04-04.
 22. ^ [a](#) [b](#) [c](#) "China to cut 90 rare earth firms into 20" [↗](#). *globaltimes.cn*. 9 September 2010.
 23. ^ [a](#) [b](#) [c](#) [d](#) [e](#) [f](#) Bontron, Cécile (2012-08-07). "Rare-earth mining in China comes at a heavy cost for local villages" [↗](#). *the Guardian*. Retrieved 2018-04-04.
 24. ^ Lee, Jason C.K.; Wen, Zongguo (2017-10-01). "Rare Earths from Mines to Metals: Comparing Environmental Impacts from China's Main Production Pathways" [↗](#). *Journal of Industrial Ecology*. **21** (5): 1277–1290. doi:10.1111/jiec.12491 [↗](#). ISSN 1530-9290 [↗](#).

External links [[edit](#)]

- [Chinese Society of Rare Earths \(CSRE\)](#) (in English) (in Chinese)

- [News about Rare earth industry in China](#) 

Categories: [Industry in China](#)

This page was last edited on 26 February 2019, at 09:22 (UTC).

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Cookie statement](#)

[Mobile view](#)



ASICs and Decentralization FAQ

1 About this document.

1. What is the thesis of this document?

The main thrust of this document is that proof-of-work schemes ought to be as simple and dependent on raw computational power as possible. That is, a proof-of-work should tend toward the thermodynamic limit (see a later section for this term) as quickly and directly as possible.

2. Is Proof of Work interesting?

Not really ☺. Replacing or tweaking the proof-of-work algorithm is one of the most popular changes to Bitcoin done by copycat “alt” currencies, but it does not enable any new use cases or features for users of the currency. It is in effect a bikeshed painting¹ change, though as we will see it is not without its dangers.

3. Why did you write this document?

For two reasons:

- (a) to organize and lay out some folklore about proof-of-work which has not been written down in one place;
- (b) to answer some common questions and suggestions regarding Bitcoin’s proof-of-work and those of other currencies (which as established above, is not interesting and therefore no fun to answer repeatedly).

2 What is Proof of Work?

1. What is Proof of Work?

As it applies to Bitcoin, a *proof of work* is a computational proof that some scarce resource was consumed. Such a proof is possible because it appears that computational resources are physically bounded by available time, space and energy²

Further, a proof of work commits to some data, effectively "signing" it with some consumed resource.

2. How and why is it used?

¹See <http://bikeshed.com/> for a history of this phrase.

²See, for example, <http://arxiv.org/abs/quant-ph/0502072> for a discussion on the emergence of fundamental computational limits from physics.

Such proofs are useful because they ensure malicious actors are limited in their capacity to produce valid proofs of alternate histories, giving the remaining actors sufficient time and bandwidth to come to consensus on a single history, as long as they have more resources than the malicious ones.

Bitcoin's main innovation was aligning economic incentives to discourage resource-rich actors from becoming malicious.

Here a "history" is formed by a chain of proofs-of-work, each of which signs (a) a collection of valid Bitcoin transactions, and (b) the previous proof-of-work. Users are able to achieve consensus not only of the transactions, but of their order in time.

3. **How does the proof-of-work affect consensus?**

In order that all actors (including those not "mining" or generating proofs of work) can quickly reach consensus, the proof-of-work must be quickly verifiable. The greater the ratio of generation time to verification time, the better.

In order that all actors can reach consensus, the proof-of-work must require enough time that previously proven transaction history can propagate and be verified before the history is extended. Here "enough time" depends on both on the amount of data to be verified (the blocksize) as well as the speed at which the proof-of-work itself can be verified.

4. **How does the proof-of-work affect decentralization?**

In order that individual actors do not gain a disproportionate advantage — so that in the economic limit one actor has all the mining power and leases it out, obviating the proof-of-work since this is functionally identical (but ecologically trillions of trillions of times worse) to this actor simply digitally signing each transaction — the proof-of-work must be completely parallelizable and require no state. That is, one actor with $2N$ hashing power should have the same return as two actors each with N hashing power.

Bitcoin achieves this property by using a proof-of-work based on finding partial hash preimages. The work is accomplished by repeated hashing, and each attempt has a tiny i.i.d. chance of success (this is derived from the so-called *random oracle assumption* of the specific hash algorithm). This gives rise to a Poisson process, which is well-understood but in some ways unintuitive.

Another crucial way that the proof-of-work affects decentralization is in its physical attributes, which we cover in the next section.

5. **Why does the Poisson process matter?**

This is important enough to warrant its own question. To avoid centralization, it is important that miners can join or leave the network at any time without penalty. It is also important that miners with a lot of computational power should not achieve a disproportionate benefit.

These two requirements are related, because they effectively say that there should be no distinction between devoting more time to mining or devoting more computational power to mining.

To guarantee this, the computation of proof-of-work must be *progress free*, that is, the proof-of-work calculation at time T should not depend on any part of a calculation at time $T' < T$. This, along with the fact that successful proof-of-work calculation should be rare enough to limit block speed, implies that the probability of a proof-of-work being calculated in a time interval $[t_0, t_1]$ is proportional to $(t_1 - t_0)$ but independent of t_0 and t_1 .

These conditions are sufficient to give rise to a *Poisson process*³.

As an example of an unintuitive behaviour of Poisson processes, the expected time right now to the next Bitcoin block is ten minutes (plus or minus the difference between actual hashpower and the difficulty-targeted hashpower). This is independent of how long ago the last block was found, even though, on average, blocks are found every ten minutes.

6. What other algorithmic considerations are there?

Two additional requirements, from Greg Maxwell, are:

- (a) The proof-of-work must be *optimization free*; that is, there should not be any algorithmic speedups which would give an advantage over the standard algorithm. If a speedup exists and is found, there is strong motivation for the discoverer to use it secretly rather than publishing it, gaining an unfair advantage. This contributes to centralization.
- (b) The proof-of-work must be *approximation free*; that is, there should not be a defective variant of the proof-of-work which achieves a speedup in excess of its defect rate. (If this is done in software, it is a special case of the above; however it can be done in hardware as well *e.g.* by using a bad multiplexer which cannot demux certain bitstrings.)

3 The physics of Proof-of-Work.

1. What is (ir)reversible computing and why does it matter?

As this is not a physics paper, this section necessarily contains several claims with neither detail nor justification. The curious reader is encouraged to read the Wikipedia article⁴ and its references.

Reversible, or adiabatic computing, is a term for computing without increase in entropy. Such computations are reversible in time and therefore need to be injective as functions of their input, so a hash-based proof-of-work cannot be completely reversible. However, some components of the hash function may be reversibly computable, which is useful because reversible computations require no energy — more concretely, in the limit as computational speed goes to zero, the energy requirement of reversible computations also goes to zero.

Ordinary, non-reversible computations are subject to Landauer's principle⁵ which provides a lower bound on the energy required to flip a single bit.

³See https://en.wikipedia.org/wiki/Poisson_process.

⁴https://en.wikipedia.org/wiki/Reversible_computing

⁵https://en.wikipedia.org/wiki/Landauer%27s_principle

What this means is that ultimately, any irreversible proof of computational work is also a proof of physical work, *i.e.* energy dissipation.

2. What is the thermodynamic limit?

The thermodynamic limit describes the state at which proof-of-work is actually done at Landauer's lower bound for required energy dissipation. At this point, we are actually better off than the requirement that a single $2N$ -powered actor has the same advantage as two N -powered actors, because now N is proportional not only to hashing speed but also to heat dissipation requirements, and it is easier for two physically-separated actors to dissipate heat than for just one.

Therefore, in the thermodynamic limit we have a physical incentive for decentralization.

Fortunately, because the thermodynamic limit is also the limit of energy efficiency, Bitcoin's incentives for efficient proof-of-work are also incentives for miners to push toward the thermodynamic limit.

3. Who cares about heat dissipation? There are massive economies of scale in buying electricity.

A frequent claim in this document is that using thermodynamic limits to obtain a proof of work creates incentives against centralization because there are diseconomies of scale in cooling, since for any mining operation there are limits to (a) how much heat can be effectively dissipated and (b) how much waste heat can be used for other purposes. To increase heat dissipation capacity requires an engineering and construction outlay; as long as waste heat is useful mining is effectively free, while past that point it is not.

However, no matter what the economics of cooling are (and the above claims are hardly uncontroversial), several people have contacted me, most notably Tim Swanson⁶ and phantomcircuit⁷ to point out that there are massive economies of scale in electrical production, enough to dwarf any decentralization caused by cooling.

I have three responses to this:

- (a) First, these problems are made worse by using other physical limits (e.g. speed-of-light, as in a memory-hard PoW) which don't require miners to cool their equipment. So if Bitcoin's economics force centralization, it is an open problem what (if anything) doesn't.
- (b) In places where the waste heat is directly useful, the cost of mining is merely the difference between electric heat production and ordinary heat production (here in BC, this would be natural gas). Then electricity is effectively cheap even if not actually cheap. As capital costs come down (as ASIC manufacturers stop being backlogged) this will become a more important factor.

⁶Private communication, largely based on data published at <http://bitcoinmagazine.com/12914/bitcoins-made-in-china/>.

⁷On #bitcoin, an unlogged IRC channel.

- (c) The main beneficiary of cheap-electricity centralization are a few well-connected businesses in China. This is something of an accident of history. Over the next few decades we will likely see this countered by
 - i. China's increasing wealth causing their regulatory and environmental costs to come in line with western nations; there will also be pressure against this sort of monopolistic behaviour.
 - ii. Highly centralized power production is inefficient because of line losses. As technology, such as solar power, matures, it will be possible to produce power more locally with lower startup costs and less dependence on geopolitical factors.

These comments reflect a very long-term view of Bitcoin; before Bitcoin can achieve its goals as a worldwide stateless currency where decentralization really matters, there are more pressing problems (usability, scalability, privacy) which need to be dealt with. My expectation is that by the time these problems are solved, these comments will be more immediate.

4 Actual frequently asked questions.

1. Are ASIC's evil?

No, dedicated hardware brings us closer to the thermodynamic limit, and is therefore eventually a good thing for mining decentralization. Also, because ASIC's produce more hashes for the same amount of energy, they produce stronger proofs-of-work with proportionally less environmental impact.

However, ASIC's bring with them a risk of manufacturer centralization, such as what we saw with Bitcoin in the early days of ASIC mining. Market forces eventually broke this monopoly⁸, and one thing which sped up the process is that Bitcoin uses the SHA2 hashing algorithm, which was designed for easy development of dedicated hardware. Therefore, relatively little startup capital is needed to develop Bitcoin ASIC's.

Further, regardless of one's personal feeling toward ASIC's, they are inevitable. Dedicated hardware will always be more efficient than general-purpose hardware (exactly because it is closer to the thermodynamic limit) and Bitcoin's incentives are aligned for ever-increasing efficiency.

2. Is ASIC resistance desirable?

No. ASIC resistance typically involves increasing algorithmic complexity to discourage ASIC developers. However, ASIC's are still inevitable; all ASIC resistance does is increase the startup capital required and therefore increase centralization of manufacturing.

⁸Well, this is a bit of a disingenuous statement. While there are many ASIC manufacturers on the market, these firms design IC's and outsource the actual IC fabrication. IC manufacturing is an extremely (and increasingly so) centralized industry. Thanks to Peter Todd for this observation.

Further, increasing the complexity of proof generation often means also increasing the complexity of proof validation, often disproportionately slow. This discourages (unpaid) non-mining validators, which also increases centralization.

3. **Is ASIC resistance possible?**

ASIC resistance, in the sense of making life difficult for ASIC manufacturers (and therefore reducing the number of distinct manufacturers) is possible. But it is impossible to create an algorithm which runs at the same speed on general-purpose and dedicated hardware (since general-purpose hardware contains many extraneous features, e.g. communication buses for peripherals), and so ultimately ASIC resistance is futile.

(Schemes such as “the developers will just change the proof-of-work algorithm if ASIC’s appear” do not even make sense — in a decentralized currency the developers have no such power, while in a centralized currency proof-of-work is a completely unnecessary waste of power.)

4. **Is memory hardness desirable?**

No. Memory hardness has the effect of increasing ASIC board footprint, weakening the heat-dissipation decentralization provided by the thermodynamic limit. Further, it increases the capital costs of mining equipment relative to the energy costs, which also encourages centralization (since established miners have amortized their equipment more than new miners). These effects are amplified by the fact that SRAM is both several times faster and several times more expensive than DRAM.

Also, memory hard proofs-of-work often require lots of memory on the part of the verifiers⁹, which is bad for decentralization as already discussed.

As an aside, since memory is far away and expensive to access on general purpose computers, memory hardness actually increases the benefit provided by ASIC’s! This is contrary to the goals of most memory-hard advocates, and as we have seen above, memory-hardness worsens the centralizing effects of ASIC’s while weakening the decentralizing effects¹⁰.

One more thing worth mentioning is time-memory tradeoff (TMTO). This is a property of an algorithm which allows higher memory usage to be traded for a heavier computational load. An algorithm which is highly susceptible to TMTO has poorly defined memory hardness, which at the very least complicates analysis. It may also cause an algorithm to fail to be optimization free.

5. **Is scrypt better than SHA2?**

No, for a few reasons:

- (a) It is much slower to validate, reducing scalability and discouraging non-mining validation.

⁹Though not necessarily — see for example John Tromp’s “cuckoo cycle” scheme at <https://bitcointalk.org/index.php?topic=451177.0>.

¹⁰In fact, at the time of this writing, scrypt ASIC’s are just appearing on the market, and do indeed provide a proportionally greater gain in hashpower than did SHA2 ASIC’s.

- (b) It is a more complex “ASIC-resistant” algorithm. See above.
- (c) It is more memory-hard than SHA2 (though depending on parameters, it can hardly be considered “memory-hard” — see Litecoin’s settings, for example). See above.

6. **Is Primecoin better than SHA2?** No, for a few reasons:

- (a) It is much slower to validate, reducing scalability and discouraging non-mining validation.
- (b) It is a more complex “ASIC-resistant” algorithm. See above.
- (c) It is an ad-hoc algorithm with no reason to believe that it is progress-free, approximation-free or optimization-free.

7. **What about “useful” proofs-of-work?**

These are typically bad ideas for all the same reasons that Primecoin is, and also bad for a new reason: from the network’s perspective, the purpose of mining is to secure the currency, but from the miner’s perspective, the purpose of mining is to gain the block reward. These two motivations complement each other, since a block reward is worth more in a secure currency than in a sham one, so the miner is incentivized to secure the network rather than attacking it.

However, if the miner is motivated not by the block reward, but by some social or scientific purpose related to the proof-of-work evaluation, then these incentives are no longer aligned (and may in fact be opposed, if the miner wants to discourage others from encroaching on his work), weakening the security of the network.

5 Proof of Stake

1. **What is Proof of Stake?**

Proof of stake is the idea that rather than consuming some physical resource, perhaps miners should consume the cryptocurrency itself, thus “bootstrapping” the security of the system from its own value, rather than requiring expensive and energy-intensive mining operations.

However, it does not appear that there is a viable way to achieve consensus through proof of stake, due to the so-called “nothing at stake” problem¹¹.

2. **In principle, how does Proof of Stake work?**

The typical way that proof of stake is accomplished is by associating “votes” to individual coins, and to require blocks to be voted highly in order to be valid. To improve scalability and prevent “tyranny of the rich” scenarios, each block is voted on by a small set of coin holders, who are determined as part of the currency’s consensus algorithm.

3. **What is the “nothing at stake” problem?**

¹¹Thanks to Andrew Miller for this name.

The problem here is that ultimately, there is no cost for users to vote on multiple forks of the blockchain, but there is some benefit. Therefore, the idea that miners will behave honestly to protect their own coins' value is actually a tragedy of the commons.

The benefit of mining on multiple forks is that the blocks themselves determine the distribution of coins and therefore the distribution of future votes. In fact, the exact voters of future blocks must be deterministic, perhaps chosen by a PRF on the contents of past blocks, so by choosing which blocks to vote on, a miner can skew the distribution of future votes in his favour. As an extreme case, the miner could simply do a proof-of-work, where the "work" is finding new blocks which give him every future vote.

This is fatal for centralization, and because each miner is individually incentivized to act this way, it is also fatal for consensus.

4. Is Proof of Stake good for anything?

Certainly. The ability to cryptographically prove one's stake in a system is an exciting feature which is unique to cryptographic goods. It can be used to construct new and innovative protocols which I have no understanding of ☺. It just can't be used to create distributed consensus.



Here's How One Japanese Newspaper Is Moving Robo-Journalism Forward

by Tim Hornyak



A man reads the Japanese newspaper Nihon Keizai Shimbun – which was not assembled by robots – in Tokyo. (YOSHIKAZU TSUNO/AFP/Getty Images)

A version of this piece appeared at the [Splice Newsroom](#).

In another step forward for robo-journalism, a regional newspaper in Japan is rolling out an artificial intelligence system that automatically generates summaries of news articles for distribution across a range of media platforms.

“With the freed-up time, AP journalists are able to engage with more user-generated content, develop multimedia reports, pursue

investigative work and focus on more complex stories.” -AP's Lisa Gibbs

The Shinano Mainichi Shimbun teamed up with Fujitsu, Japan's largest IT services company, to create the software based on technology developed by Fujitsu Laboratories. Staff at the broadsheet have been producing summaries manually, a task that takes up to five minutes per article. The software creates summaries instantly and with greater accuracy than a different summarizing method that begins with the lead and stops when the word limit is reached, according to Fujitsu:

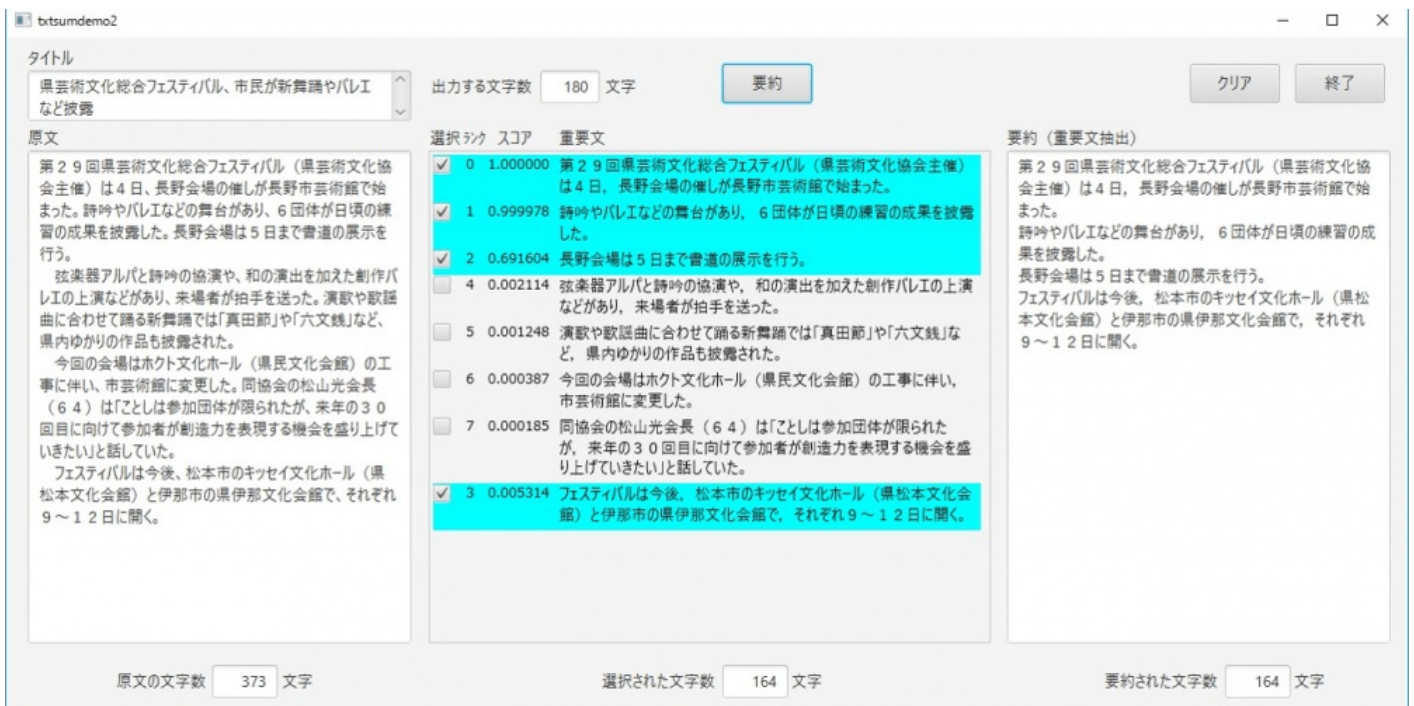
ADVERTISEMENT

“The system uses a combination of natural language processing and machine learning to pick out the most salient parts of the article, scoring each sentence in terms of importance.”

During a trial, it was trained on a dataset of 2,500 articles from the newspaper as well as their manually compiled summaries.

“By pairing the original articles with the summaries and defining that as reference, or teacher data, we built an ‘important sentence extraction model’ that evaluates the content importance according to individual sentences, as well as a ‘sentence-shortening model’ that maintains sentence structure while deleting unnecessary words,” says Masato Yokota, a director at Fujitsu’s State Infrastructure and Finance Business Group.

The software can work with articles written in Japanese or English. It was built with a web API that can be easily inserted into the existing editorial workflow. A “summary” button activating the API was implemented into the editing screen for the paper’s cable TV news, Yokota said.



A screenshot of the AI system from its trial period shows the original article in Japanese (left), an automatically generated ranking of sentences by importance (center), and the summarized text (right). (Courtesy the Splice Newsroom)

Robots Vs. Journalists

First published in 1873, the Shinano Mainichi Shimbun is one of Japan’s oldest dailies. Headquartered in Nagano, northwest of Tokyo, it claims a morning-edition circulation of 487,000 copies and distribution to 61% of households in Nagano Prefecture.

“The third-wave AI is set to become a trend of great relevance, and now is the time to make concerted efforts in improving the newspaper production workflow as well,” says Hiroshi Misawa, the paper’s managing director.

The Shinmai, as it’s known, plans to roll out the system in April for its cable TV news summary service, with an eye to speeding up news updates.

The summarizing AI joins a host of other automated news applications sometimes described as automated or augmented journalism. Heliograf, the Washington Post’s own news bot, produced about 300 briefs on the Rio Olympics of 2016, and has since covered U.S. elections and high school football games; it produced about 850 articles in its first year, according to [Digiday](#). The Associated Press worked with AI firm [Automated Insights](#) to deploy software to cover earnings reports.



The Fujitsu pavillon at the Mobile World Congress 2016. (Manuel Blondeau/ AOP.Press/Corbis via Getty Images)

“Through automation, AP is providing customers with 12 times the corporate earnings stories as before (to over 3,700), including for a lot of very small companies that never received much attention,” AP global business editor Lisa Gibbs was quoted as saying in [a 2017 report](#).

“With the freed-up time, AP journalists are able to engage with more user-generated content, develop multimedia reports, pursue investigative work and focus on more complex stories.”

Tim Hornyak is a freelance journalist based in Tokyo. He is the author of [Loving the Machine: The Art and Science of Japanese Robots](#).

Related Stories

AI is Changing These Newsrooms: What It Means for Digital Publishing

How Trinity Mirror's In-House Analytics Platform Helps Journalists

Heads Up! Marketers to Combine AI with 44 Billion Cameras

Explore MediaShift

Why Schools Should Stop Banning Cell Phones, and Use Them for Learning

Who Gets The Most Traffic Among Conservative Websites?

Cell Phones in Classrooms? No! Students Need to Pay Attention

Tagged:

ai

ap

Fujitsu

Heliograf

japan

robo-journalism

robot journalism

washington
post

MEDIASHIFT

Metrics

EducationShift

Social Media

Business

Global View

About

Contact Us

Sponsorships

MediaShift Events

DigitalEd



With your host
Mark Glaser

[Privacy Policy](#) [Terms of Use](#)
Copyright MediaShift
2016

ISSN 1551-3483



9 771551 348002



<https://scale.qihardware.org>