

# Issue 058: Community

Adrian Kosmaczewski

July 3<sup>rd</sup>, 2023



Welcome to the fifty-eighth issue of *De Programmatica Ipsum*, about *Community*.

In this edition:

- We enumerate the ways in which a technical community can degrade<sup>1</sup>.
- In the Library section<sup>2</sup>, we review “Working in Public” by Nadia Asparouhova<sup>3</sup>.
- In our Vidéothèque section<sup>4</sup>, we learn about technical marketing from Guy Kawasaki<sup>5</sup>.

We would also like to thank our patrons who generously contribute every month (or have contributed in the past) to our work and help us run this magazine. Thank you so much! In alphabetical order: Adam Guest, Adrian Tineo Cabello, Benjamin Sheldon, Christopher Nascone, Jean-Paul de Vooght, Patryk Matuszewski, Paul Hudson, Quico Moya, Roger Turner, and Szymon Licau.

Enjoy this issue! Please subscribe to our free newsletter<sup>6</sup> to stay updated about new releases, share the articles on social media, or contribute<sup>7</sup> if you would like to support our work.

<sup>1</sup><https://deprogrammaticaipsum.com/the-tragedy-of-the-common-enemy/>

<sup>2</sup><https://deprogrammaticaipsum.com/category/library/>

<sup>3</sup><https://deprogrammaticaipsum.com/nadia-asparouhova/>

<sup>4</sup><https://deprogrammaticaipsum.com/category/videotheque/>

<sup>5</sup><https://deprogrammaticaipsum.com/guy-kawasaki/>

<sup>6</sup><https://deprogrammaticaipsum.com/newsletter/>

<sup>7</sup><https://deprogrammaticaipsum.com/contribute/>

Cover photo by Mike Erskine<sup>8</sup> on Unsplash<sup>9</sup>.

---

<sup>8</sup>[https://unsplash.com/@mikejerskine?utm\\_source=unsplash&utm\\_medium=referral&utm\\_content=creditCopyText](https://unsplash.com/@mikejerskine?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText)

<sup>9</sup>[https://unsplash.com/photos/Xtnt5xtK03E?utm\\_source=unsplash&utm\\_medium=referral&utm\\_content=creditCopyText](https://unsplash.com/photos/Xtnt5xtK03E?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText)

# The Tragedy Of The Common Enemy

Adrian Kosmaczewski

July 3<sup>rd</sup>, 2023



My father, whom I have already mentioned in this magazine in a previous article about skeuomorphism<sup>1</sup>, is an architect and urbanist. While walking down the streets of Buenos Aires, he would often mention what he thinks is one of the greatest paradoxes of our time: humans have invented cities to feel safe from predators and to ensure their survival, yet most cities are nowadays among the unsafest places for humans to live.

Software developers are, to a large extent, still humans, although this situation might change substantially between this and the next decade. As such, we will still gather around our favorite technologies, clustering as if we were peasants migrating to the city, or stars orbiting around a galaxy<sup>2</sup>, and participating (or not) in the various rituals that make up those communities. There are many different activities possible in such cities and galaxies: attending conferences and meetups, publishing blog posts or videos, and of course, contributing code to our favorite open-source<sup>3</sup> projects.

25 years after the term “Open-Source” has been coined<sup>4</sup>, and almost 40 years after the creation of the Free Software Foundation<sup>5</sup>, economists are still scratching their heads, watching one of the wealthiest segments of society work for free during their leisure time, towards the building of another tragedy of the commons. The reasons why they do this are beyond the scope of this article; let us ask ourselves, instead, what kind of communities spring up to life around those projects.

<sup>1</sup><https://deprogrammaticaipsum.com/the-button-and-the-spoon/>

<sup>2</sup><https://akos.ma/blog/the-developer-guide-to-migrate-across-galaxies/>

<sup>3</sup><https://deprogrammaticaipsum.com/issue-21-open-source/>

<sup>4</sup><https://opensource.com/article/18/2/coining-term-open-source-software>

<sup>5</sup>[https://en.wikipedia.org/wiki/Free\\_Software\\_Foundation](https://en.wikipedia.org/wiki/Free_Software_Foundation)

Many of those communities present themselves as built around positive values. Let us look at their websites. For Ruby on Rails<sup>6</sup>, the choice of words says it all: “improving”, “help”, “thanks”, and even a heart emoji at the bottom. The OpenShift Commons<sup>7</sup> website shows various ways to “get involved” in the community. The F# community<sup>8</sup> has the literal word “community” on its URL, and features a big map with lots of local user groups scattered around the world. The WordPress project, having recently celebrated its 20th anniversary, is happy to share another map<sup>9</sup> showing the distribution of its community members around planet Earth. The Cloud Native Computing Foundation sponsors a series of events called the “Kubernetes Community Days” spread across all continents, one of which recently took place in Zürich<sup>10</sup>.

Debian<sup>11</sup> goes even further, screaming out loud the words “The Debian Project *is* a Community”:

Everyone can be a part of our community; you don’t have to be a developer or sysadmin. Debian has a democratic governance structure. Since all members of the Debian project have equal rights, Debian cannot be controlled by a single company. Our developers are from more than 60 different countries, and Debian itself is translated into more than 80 languages.

Some projects even have “community architects” in place, like CentOS or Fedora<sup>12</sup>, whose role is to ensure that the interactions between those “community” members are civilized, or at least aligned to whatever agenda their major funding members would like to push.

All very positive instead.

But let us be honest: nothing binds people together better than a common enemy. Negative values have always been a strong glue to build communities, arguably stronger than positive values. Humans will be humans.

Examples abound in the computer world; haters of systemd<sup>13</sup> get together and build their own Linux distributions, like Devuan<sup>14</sup>, Obarun<sup>15</sup>, Artix<sup>16</sup>, or Void Linux<sup>17</sup>. Haters of bureaucracy and processes in the Rust language get together and build Crablang<sup>18</sup>. Haters of Red Hat gather around AlmaLinux<sup>19</sup> and Rocky<sup>20</sup>. Haters of corporations mishandling their communities revolt (usually due to a clumsy-slash-evil CEO, or an upcoming IPO), like in the recent cases of Stack Exchange<sup>21</sup> or Reddit<sup>22</sup>.

These examples point out to the major risk faced by open-source communities: forking.

---

<sup>6</sup><https://rubyonrails.org/community>

<sup>7</sup><https://commons.openshift.org/>

<sup>8</sup>[https://community.fsharp.org/user\\_groups](https://community.fsharp.org/user_groups)

<sup>9</sup><https://thewp.world/>

<sup>10</sup><https://kcdzurich.ch/>

<sup>11</sup>[https://www.debian.org/intro/why\\_debian](https://www.debian.org/intro/why_debian)

<sup>12</sup><https://www.youtube.com/watch?v=JaZAB2VwnzU>

<sup>13</sup><https://en.wikipedia.org/wiki/Systemd>

<sup>14</sup><https://www.devuan.org/>

<sup>15</sup><https://web.obarun.org/>

<sup>16</sup><https://artixlinux.org/>

<sup>17</sup><https://voidlinux.org/>

<sup>18</sup><https://crablang.org/>

<sup>19</sup><https://almalinux.org/blog/impact-of-rhel-changes/>

<sup>20</sup><https://rockylinux.org/news/2023-06-22-press-release/>

<sup>21</sup><https://devclass.com/2023/06/12/stack-overflow-strives-to-protect-community-content-from-ai-firms-striking-mods-say-re-enable-the-data-dumps/>

<sup>22</sup><https://www.theverge.com/23768252/reddit-blackout-moderation-lessons-elon-musk>

Thus were born LibreOffice (itself forked from OpenOffice), MariaDB (from MySQL), OpenSSL (from SSLeay<sup>23</sup>), and even WordPress (a fork of b2/cafelog). And we are not even mentioning a myriad of BSD kernel variants.

Surprisingly enough, some projects have not had major forks in decades: Vim, Emacs, Git, or Linux. The last two managed by the same person, somebody who is not particularly well-known for his attention to people's sensibilities. The same can be said about David Heinemeier Hansson, and other "benevolent dictators" in other projects, who can passive-aggressively drive their projects to where they believe should go.

Much is debated nowadays (in rather derogatory terms) about the "wokeness" (the use of quotes is to highlight the derogatory part) and "inclusivity" factor in online communities. Many open-source communities are, as a matter of fact, strictly unsafe for non-Caucasian males between 25 and 35 years old. And I am not even talking about "rude" behavior (comes to mind the infamous "RTFM or GTFO" or "use the source, Luke" attitudes), but openly hostile acts of incivility<sup>24</sup> that, should they happen in the "real" world, would face immediate reaction from the powers that be.

Online communities, however, are a new far west for many predatory acts.

The spread of savage capitalism practices has made us forget that the primary reason we gather is to *collaborate*. We created cities so that we would have easier access to food and shelter. The idea was to help one another. The idea was not to use one another, but to give a helping hand whenever we could.

Code these days is free as in freedom, and very often free as in beer. We need more than ever to remember that kindness is the killing feature of our species. Be kind to one another, at all times. Take care of one another. Do not bully, harass, tamper, harm, or belittle anyone, for our human existence is already quite complex as it is.

People seem to really have bought into the capitalist version of open source where software is still a product that requires support and marketing and a roadmap and exists to serve a user community separate and apart from the project.

But a whole lot of open source is really just a sharing economy. It's devs doing something they found useful and deciding to share it rather than hoard it. Those devs don't owe anyone extra labor just because they chose to share.

(Source<sup>25</sup>)

To all of you managing open-source projects, here is our call to action: do not tolerate intolerance<sup>26</sup>. Manage your "degree of openness"<sup>27</sup> if needed; you could go full Open-Source by Default<sup>28</sup>, as Orta Therox<sup>29</sup> once said, but you can also choose whichever way you would like to do it.

To all of you joining open-source projects, please make sure to do your due diligence, despite all the enthusiasm you might have to join a particular community. Read the forums, review

---

<sup>23</sup><https://en.wikipedia.org/wiki/SSLeay>

<sup>24</sup><https://medium.com/@imashadowphantom/mariadb-com-is-dead-long-live-mariadb-org-b8a0ca50a637>

<sup>25</sup><https://hackers.town/@calcifer/110531948483085056>

<sup>26</sup>[https://en.wikipedia.org/wiki/Paradox\\_of\\_tolerance](https://en.wikipedia.org/wiki/Paradox_of_tolerance)

<sup>27</sup><https://github.com/readme/featured/how-open-is-open-source>

<sup>28</sup><https://artsy.github.io/series/open-source-by-default/>

<sup>29</sup><https://github.com/orta>

old pull requests, and take a conscious decision: is this a community you want to be a part of?

As my father pointed out at the beginning of this article, most modern cities on Earth have become unlivable swamps of dirt, crime, corruption, isolation, and despair. As humans, we have transformed pre-capitalist cities, which used to be our primary social vehicle of survival and collaboration, into one of techno-neoliberalism, destruction, and desolation. Communities built around open-source projects run the same risk, every single day. Let us not let this happen.

Paraphrasing Alfred in the DC Cinematic Universe, “We are not our enemy.”

And, who knows. Maybe one day, after we have conquered code<sup>30</sup>, we will apply what we have learned by giving away our free labor towards the creation of other common goods, in the real world. Think about it; spending 3 hours a day, but instead of coding, feeding homeless people, planting trees, or helping the elders.

PS: readers of this magazine shape a small yet thriving community, built on top of decidedly positive values. Among the most important, we can mention: opening the eyes of our readers to the history of computing and programming; to highlight software developers’ role as a major shaping force for society; and to increase the overall awareness of professionals in our industry. And maybe even to make you smile from time to time. Some of those community members even provide financial support<sup>31</sup> for this operation to sustain itself, and we could not be happier to have you on board.

Cover photo by Jonathan Harrison<sup>32</sup> on Unsplash<sup>33</sup>.

---

<sup>30</sup><https://deprogrammaticaipsum.com/the-conquest-of-code/>

<sup>31</sup><https://deprogrammaticaipsum.com/contribute/>

<sup>32</sup>[https://unsplash.com/@mylimbsarelong?utm\\_source=unsplash&utm\\_medium=referral&utm\\_content=creditCopyText](https://unsplash.com/@mylimbsarelong?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText)

<sup>33</sup>[https://unsplash.com/photos/G4UAIDJeJFk?utm\\_source=unsplash&utm\\_medium=referral&utm\\_content=creditCopyText](https://unsplash.com/photos/G4UAIDJeJFk?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText)

# Guy Kawasaki

Adrian Kosmaczewski

July 3<sup>rd</sup>, 2023



There is a website out there providing an answer to the question “What is Developer Relations?,”<sup>1</sup> a question that this author, whose job title is precisely that one, gets a lot. According to this resource, Developer Relations is an umbrella term encompassing three major areas of activity: “Community,” “Content,” and “Product;” a perfect description of a job role, at least as far as this author is concerned.

Communities are so important for product marketing these days, that the major book in the field of Developer Relations, “The Business Value of Developer Relations”<sup>2</sup> by Mary Thengvall, focuses almost entirely on the subject of community management. Part one is about describing the value of the community, and making a business case out of it (a much-needed step in traditional corporate structures); part two is about building and maintaining it. Case in point, during a recent CentOS podcast episode<sup>3</sup> hosted by Shaun McCance, Justin Flory, the “Community Architect” for the Fedora project acquiesced, stating that his job essentially consisted of community management.

The size of those communities can be measured: according to page 26 of the 2022 State of Developer Relations Report<sup>4</sup>, 30% of respondents report that they actively manage communities from 100 to 5000 members, with another 10% managing humongous communities with over 2 million members. Of course, not all members are active, with an average of 35% of active users.

Technical communities provide software businesses with an audience, a test bed, and eventually, a customer pool for their products and services, but this only works if the products are good enough to begin with. This insight was clearly defined by Guy Kawasaki<sup>5</sup>, arguably the

---

<sup>1</sup><https://www.whatisdevrel.com/>

<sup>2</sup><https://www.persea-consulting.com/book>

<sup>3</sup><https://www.youtube.com/watch?v=JaZAB2VwnzU>

<sup>4</sup><https://www.stateofdeveloperrelations.com/>

<sup>5</sup>[https://en.wikipedia.org/wiki/Guy\\_Kawasaki](https://en.wikipedia.org/wiki/Guy_Kawasaki)

person who invented the field of Developer Relations, during his tenure as Chief Evangelist at Apple from 1983 to 1987.

Please pay attention to the word “Evangelist” in the job title. As both Guy Kawasaki and Wikipedia<sup>6</sup> correctly explain,

The word evangelist comes from the Koine Greek word εὐαγγέλιον (transliterated as euangelion) via Latinised evangelium as used in the canonical titles of the Four Gospels, authored by (or attributed to) Matthew, Mark, Luke, and John (also known as the Four Evangelists). The Greek word εὐαγγέλιον originally meant a reward given to the messenger for good news (εὖ = “good”, ἀγγέλλω = “I bring a message”; the word “angel” comes from the same root) and later “good news” itself.

He wrote about his experience at Apple in the 1990 book “The Macintosh Way”<sup>7</sup>, now freely available<sup>8</sup> to download. In chapter 6 of this book, Mr. Kawasaki literally refers to some audience segments as “The Cult,” adding yet another layer of occultism to the work of technical product marketing:

There is a small part of emerging markets called The Cult. It is the “head” of the market and contains hardcore power users and aficionados, plus a small percentage of luminaries, analysts, press, and dealers. The Cult is intrinsically elite, more discerning, less tolerant of mediocrity, and more willing to accept new ideas. It categorically rejects some products as forcefully as it accepts others.

If The Cult likes your product, the rest of the market and all of the layers are more likely to follow.

To summarize, he was evangelizing to a cult. Not very reassuring, if you ask me.

In any case, this month’s Vidéothèque movie is the recording<sup>9</sup> of a 2017 Facebook Live session, where Guy Kawasaki explained his experience as a software evangelist at Apple. In this video, he points out several self-evident yet important truths about technical product marketing, the first of which is that it is easier to evangelize a great product than a crappy one. Sounds obvious, but it is always important to remind companies about this simple fact.

Evangelizing great products turns quickly into a self-fulfilling prophecy, with strong viral effects; users will start convincing others of the benefits of a great new product, without further spending by the original company. This is the Holy Grail of technical product marketing; when a technical community becomes a Tupperware party<sup>10</sup>.

The key to achieving this self-fulfilling prophecy consists in planting many seeds, and listening to those first customers; your marketing message might as well be hidden in their early feedback. Put in other words, “there is no wrong consumption of your product”, which means that the primary use intended by the product designers might not be what the market wants to do with it. Are you open-minded enough to adapt your marketing message to whatever the market is doing with your baby?

Ironically enough, on the dark side of community management, we find Apple itself. The Cupertino company is well-known among tech companies for being the one that ignores its

<sup>6</sup><https://en.wikipedia.org/wiki/Evangelism>

<sup>7</sup><https://guykawasaki.com/books/the-macintosh-way/>

<sup>8</sup><https://archive.org/details/the-macintosh-way>

<sup>9</sup><https://www.youtube.com/watch?v=Tx6m9H2Apc>

<sup>10</sup>[https://en.wikipedia.org/wiki/Tupperware#Tupperware\\_parties](https://en.wikipedia.org/wiki/Tupperware#Tupperware_parties)



community the most. This author made the bitter discovery of Apple's disdain for external communities in 2008, while co-organizing the first iPhone Conference<sup>11</sup> in Switzerland, when we received a literal "cease-and-desist" letter from Apple. Also comes to mind the Fucking NDA<sup>12</sup>; those with memory will remember. Such behavior is a relic from the Jobs-era, fueled by a furious tendency to control the message around and about their products.

This attitude is an integral part of the company's DNA, for not even during the darkest moments of the company, around 1996, did the company acknowledge the immense work done by the Macintosh User Groups. Guy, however, rightfully recognizes the role of the MUG at minute 17:00<sup>13</sup> in the video, providing a pure form of evangelism, without any support from Apple. Driven by the unshakable mantra whereby great products beget great user groups. Even nowadays, Apple communities are driven by outsiders, passionate individuals such as John Sundell<sup>14</sup> and many others, without any support whatsoever from the "mothership."

But there is far more to community management than that. If you are interested in knowing more about Tech Evangelism, check Guy's course on Udemy<sup>15</sup> and follow his Remarkable People Podcast<sup>16</sup>.

Cover snapshot chosen by the author.

---

<sup>11</sup><https://web.archive.org/web/20081011200352/https://www.iphone-conference.ch/>

<sup>12</sup><https://web.archive.org/web/20080728022219/http://www.fuckingnda.com:80/>

<sup>13</sup><https://youtu.be/Tlx6m9H2Apc?t=1020>

<sup>14</sup><https://www.swiftbysundell.com/>

<sup>15</sup><https://www.udemy.com/course/the-art-of-evangelism/>

<sup>16</sup>[https://www.youtube.com/playlist?list=PLVQWnQzWzrOM4tjZ56Fvb3\\_Iw5yitdU6s](https://www.youtube.com/playlist?list=PLVQWnQzWzrOM4tjZ56Fvb3_Iw5yitdU6s)

# Nadia Asparouhova

Adrian Kosmaczewski

July 3<sup>rd</sup>, 2023



In the twenty-five years since the appearance<sup>1</sup> of the phrase “Open Source”, many authors have tried to explain this simple fact: why do software developers willingly and spontaneously collaborate, often on a *pro bono* basis, to the creation of open-source software? And most importantly, how does this even happen? Many books have been written around this seemingly illogical fact.

Come to mind three major examples: the practical, freely available, and updated in 2020 “Producing Open Source Software: How to Run a Successful Free Software Project”<sup>2</sup> by Karl Fogel; the historical in-depth analysis of “The Success of Open Source”<sup>3</sup> (2005) by Steven Weber, political scientist at the University of California, Berkeley; “The Open Source Way”<sup>4</sup> online guidebook; and “Working in Public: The Making and Maintenance of Open Source Software”<sup>5</sup> (2020) by Nadia Eghbal, also known as Nadia Asparouhova, the subject of this month’s Library section and arguably the most important voice in the world of open-source community management and governance of this decade.

Let us talk about credentials. Nadia is a Harvard Kennedy School fellow<sup>6</sup>, a researcher specialized in digital infrastructure and community governance, and more to the point, an ex-

<sup>1</sup><https://opensource.com/article/18/2/coining-term-open-source-software>

<sup>2</sup><https://producingoss.com/>

<sup>3</sup><https://www.hup.harvard.edu/catalog.php?isbn=9780674018587>

<sup>4</sup><https://www.theopensourceway.org/>

<sup>5</sup><https://press.stripe.com/working-in-public>

<sup>6</sup><https://www.belfercenter.org/person/nadia-eghbal>

GitHub employee, as explained on her personal website<sup>7</sup>. Most importantly, she is the author of “Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure”<sup>8</sup>, a freely available ground-breaking report published with support from the Ford Foundation<sup>9</sup> in 2016, where Nadia proposes the notion that open-source code is akin to public infrastructure, highlighting the urgent need for creators to be supported in their work:

The decentralized, non-hierarchical nature of the public coding community makes it difficult to secure pay for coders, yet the work that emerges from it is the foundation for a digital capitalist economy. Increasingly, developers are using shared code without contributing to its maintenance, leaving this infrastructure strained and vulnerable to security breaches.

Her 2020 book “Working in Public” builds upon these ideas, bringing quite a few more to the table. Let us enumerate a few: source code features, in its static form, zero marginal distribution costs, but non-zero marginal maintenance costs; each new contributor adds a non-negligible burden to maintainers.

Another interesting idea: GitHub is akin to Instagram, particularly in terms of the impact of celebrity status for a few developers on the platform. Not only that, but their reputation also decreases with a “half-life” like that of a battery or a radioactive component.

And finally, the idea that open-source code is consumed as a public good, but produced as a commons, where the attention of the maintainers is the limited resource. Most importantly, she makes a clear distinction between the words “public” and “participatory;” both are commonly confused in the minds of open-source zealots, on both sides of the contributor-maintainer dichotomy.

As an example of the tone of the book, Nadia provides in chapter 2 a colorful analogy to understand the attention economy created by open-source projects: depending on their contributor and user growth patterns, said projects can be classified as either *federations*, *clubs*, *toys*, or *stadiums*. She also defines and builds upon the notions of excludable, non-excludable, rivalrous, and non-rivalrous goods, directly taken from the world of economics, providing a solid theoretical foundation underpinning her own conclusions.

Her book is filled with useful references, including screenshots of pull requests taken verbatim from GitHub, and quoting major voices in the open-source community.

She is probably the only person today that dares to mention the “money” word in the dynamics of open-source projects. Chapter 5 of her book has a full section dedicated to this, discussing both *intrinsic* and *extrinsic* motivators for software developers to create their projects and for collaborators to join said communities. This discussion involves the role of major corporations, more and more interested in funding that work, and how those contributions align with their market strategies.

Nadia Asparouhova has definitely become a voice to follow closely in the next few years. As much as developers tend to hate politics, both Steve Weber and Nadia Asparouhova have highlighted the increasingly important political perspective to understand how an open-source project is made, and even more important, maintained.

---

<sup>7</sup><https://nadia.xyz/>

<sup>8</sup><https://www.fordfoundation.org/work/learning/research-reports/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure/>

<sup>9</sup><https://www.fordfoundation.org/>

If you would like to know more about Nadia Asparouhova, subscribe to her newsletter<sup>10</sup>; check out “Where Money meets Open Source”<sup>11</sup>, a talk at JupyterCon 2017; watch “The Making and Maintenance of our Open Source Infrastructure”<sup>12</sup>, a 2020 web seminar sponsored by the Long Now Foundation<sup>13</sup>; read the Wired article<sup>14</sup> she recently wrote about GitHub’s office; and read this 2020 interview<sup>15</sup> published at The Pull Request newsletter.

The book “Working in Public: The Making and Maintenance of Open Source Software”<sup>16</sup> is available on Stripe Press, and is a required read by anyone interested in the dynamics of open-source communities today.

Cover photo by the author.

---

<sup>10</sup><https://nayafia.substack.com/>

<sup>11</sup><https://www.youtube.com/watch?v=bjAinwgyQqc>

<sup>12</sup><https://www.youtube.com/watch?v=5wLKnN3To-k>

<sup>13</sup><https://longnow.org/>

<sup>14</sup><https://www.wired.com/story/github-tech-values/>

<sup>15</sup><https://www.thepullrequest.com/p/nadia-eghbal>

<sup>16</sup><https://press.stripe.com/working-in-public>