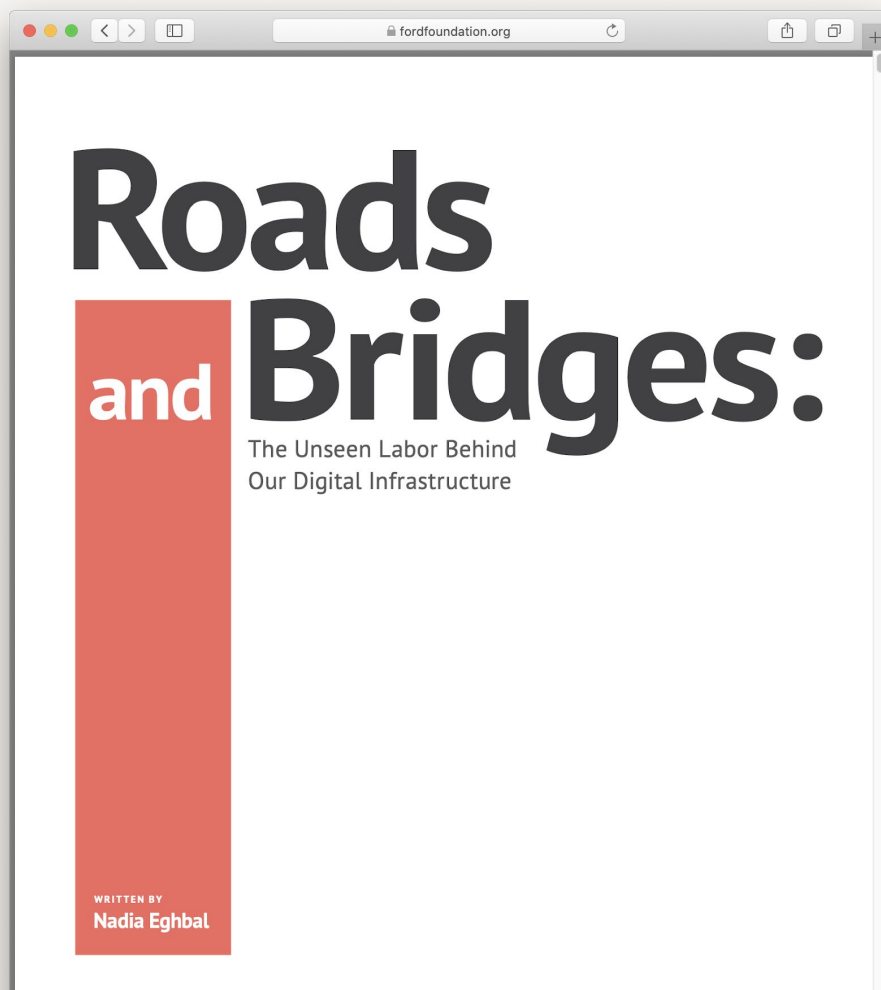




Does open source need its own Priority of Constituencies?





Tobie Langel (@tobie)
Principal, UnlockOpen

Why Open Source Failed



John Mark

Follow

Jul 30, 2018 · 10 min read ★

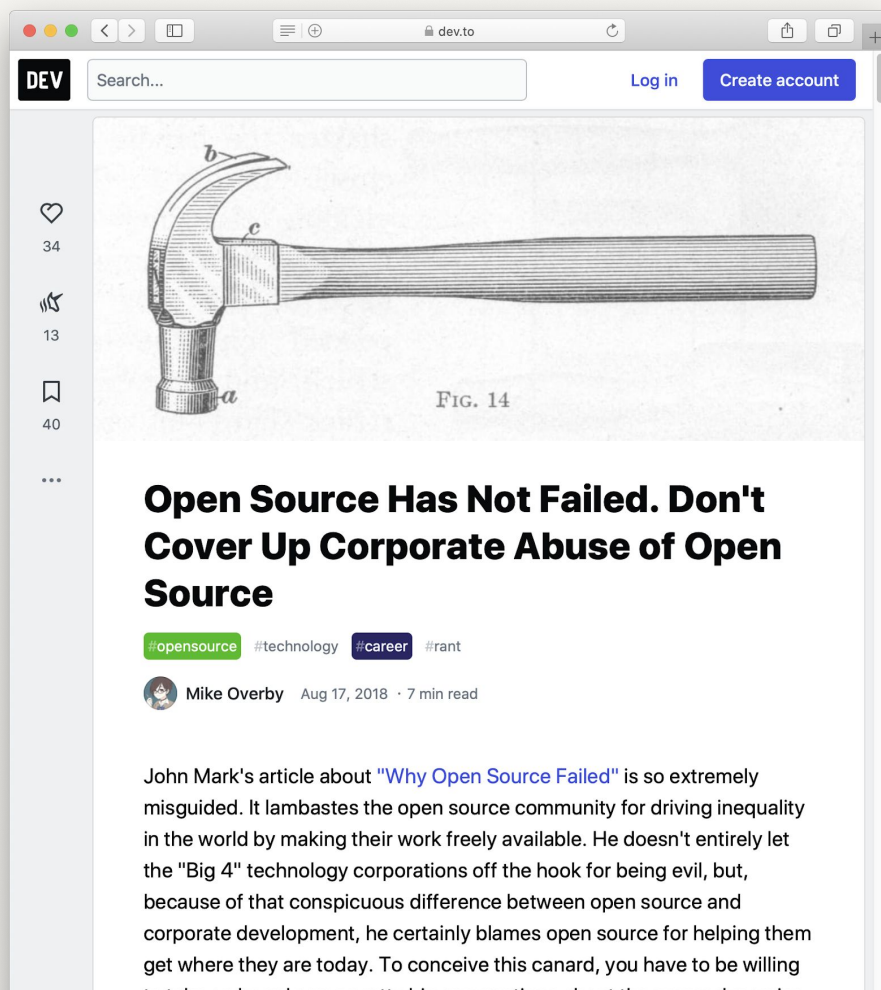


"Money" by Andrew Magill CC-BY

(If you like this essay, you may be interested in my follow-up, "[Save Open Source, Save the World](#)")

2018 is the 20th anniversary of the term "open source", and a good number

Tobie Langel (@tobie)
Principal, UnlockOpen



Featured Article

The crusade against open-source abuse



Cloud infrastructure providers threaten the viability of open source

Salil Deshpande @salil / 6:30 pm CET • November 29, 2018

Comment



Image Credits: Brad / Flickr under a CC BY 2.0 license.

aws

Contact SalesSupport▼My Account▼Create an AWS Account

ProductsSolutionsPricingDocumentationLearnPartner NetworkAWS Marketplace


C>Q

Blog HomeCategory▼Edition▼Follow▼

Search Blogs

Keeping Open Source Open – Open Distro for Elasticsearch

by Adrian Cockcroft | on 11 MAR 2019 | in [Amazon Elasticsearch Service](#), [Open Distro For Elasticsearch](#), [Open Source](#) | [Permalink](#) | [Comments](#) | [Share](#)



[中文版](#) – At AWS, we focus on solving problems for customers. Over the years, customer usage and dependencies on open source technologies have been steadily increasing; this is why we've long been committed to open source, and our pace of contributions to open source projects – both our own and others' – continues to accelerate.


When AWS launches a service based on an open source project, we are making a long-term commitment to support our customers. We contribute bug fixes, security, scalability, performance, and feature enhancements back to the community. For example, we have been a significant contributor to Apache Lucene, which powers Amazon Elasticsearch Service. The Amazon EMR team has been making contributions to the Hadoop ecosystem for many years, and the Amazon Elastic Container Service for Kubernetes (EKS) team has been contributing to Kubernetes. We also invest in open source communities, [training developers and operators](#), and sponsor open source events and conferences such as [ApacheCon](#) and [KubeCon](#), and recently

Resources

[Open Source at AWS](#)
[Projects on GitHub](#)

Follow

- [AWS Open Source](#)
- [AWS Cloud](#)
- [Facebook](#)
- [LinkedIn](#)
- [Twitch](#)
- [Open Source RSS](#)
- [Email Updates](#)

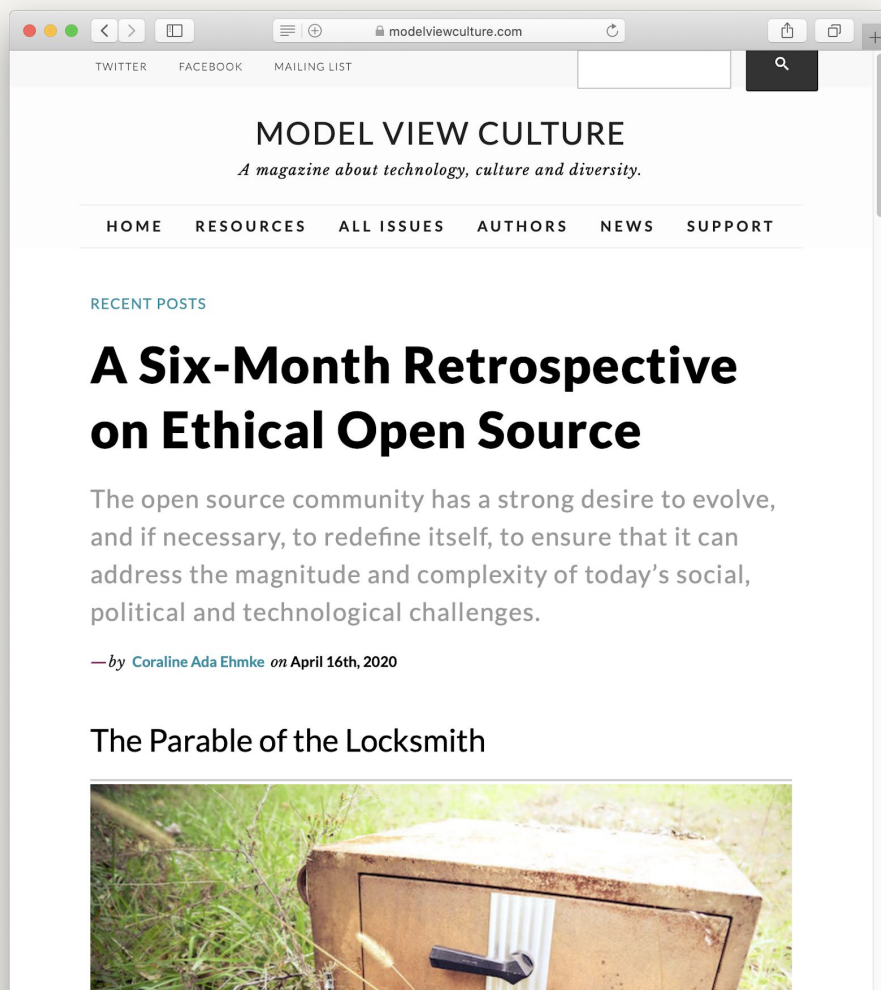


AWS Events

Discover the latest AWS events in your region

[Learn more >](#)

Tobie Langel (@tobie)
Principal, UnlockOpen



youtube.com


YouTube CH

Search

SIGN IN

What Comes After Open Source?

Bruce Perens
bruce@perens.com




0:10 / 59:44

Chat Replay is disabled for this Premiere.

What Comes After Open Source

658 views • Premiered Aug 24, 2020

29 2 SHARE SAVE ...

 **Bruce Perens**
34 subscribers

SUBSCRIBE

Free Software / Open Source have been going on for 37 years, and it's time to assess our performance and see if we can preserve our successes and improve upon our failures. I propose preserving Open Source as it exists and going forward with a new paradigm.

SHOW MORE

Tobie Langel (@tobie)
Principal, UnlockOpen

Open source crisis

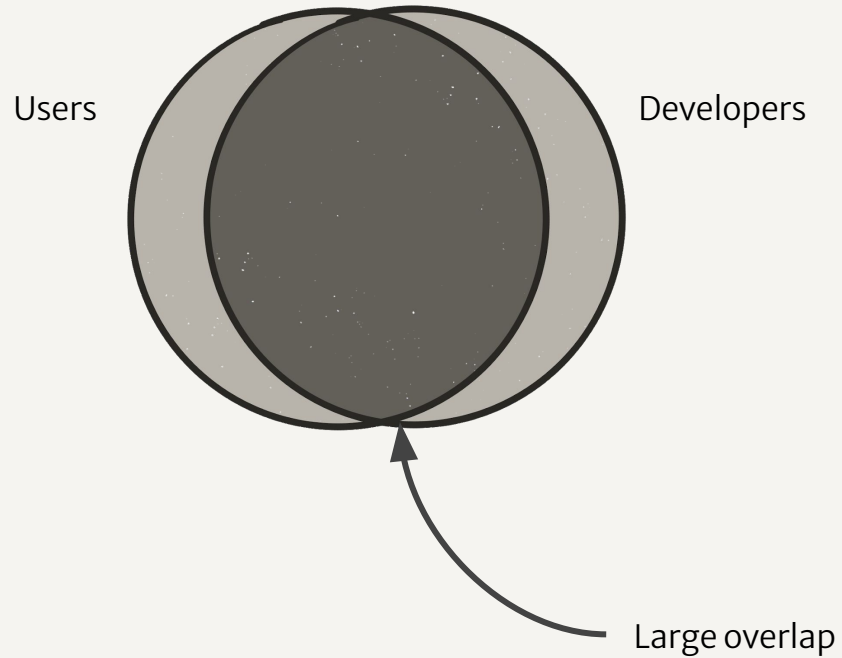
- Maintainer burnt out.
- Cloud capturing too much value.
- Ethical concerns over impact on end-users.

What's going on?

We're witnessing the emergence of new constituencies.

The 4 freedoms & the OSD

The 4 freedoms & the OSD



How do we handle this increased diversity of constituencies?

W3C's Priority of Constituencies



Tobie Langel (@tobie)
Principal, UnlockOpen

3.2. Priority of Constituencies

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

3.3. Secure By Design

Ensure that features work with the security model of the web. Preferably address security considerations directly in the specification.

Communicating between documents from different sites is useful, but an unrestricted version could put user data at risk. Cross-document messaging is designed to allow this without violating security constraints.

3.4. Separation of Concerns

HTML should allow separation of content and presentation. For this reason, markup that expresses structure is usually preferred to purely presentational markup. However, structural markup is a means to an end such as [media independence](#). Profound and detailed semantic encoding is not necessary if the end can be reached otherwise. Defining reasonable default presentation for different media may be sufficient. HTML strikes a balance between semantic expressiveness and practical usefulness. Names of elements and attributes in the markup may be pragmatic (for brevity, history, simplicity) rather than completely accurate.

The `article` element defines an individual article, but not the details of how it is displayed. A journal article may be the only article on a page, formatted in multiple columns, while a blog post may share a page with multiple other articles and be presented in a box with a border.

The `b` and `i` elements are widely used — it is better to give them good default rendering for various media including aural than to try to ban them.

3.5. DOM Consistency

The two serializations should be designed in such a way that the DOM trees produced by the respective parsers appear as consistently as feasible to scripts and other program code operating on the document trees. Discrepancies can be allowed for compatibility with legacy implementations, but the differences should be minimized.

Also, unless required for compatibility with legacy implementations and deployed content, gratuitous difference in syntactic appearance should be avoided as well.

The HTML (`text/html`) parser puts elements in the `http://www.w3.org/1999/xhtml` namespace in the DOM for compatibility with the XML syntax of HTML 5.

§ 1.1. Put user needs first (Priority of Constituencies)

If a trade-off needs to be made, always put user needs above all.

The internet is for end users: any change made to the web platform has the potential to affect vast numbers of people, and may have a profound impact on any person's life.

User needs come before the needs of web page authors, which come before than the needs of user agent implementors, which come before than the needs of specification writers, which come before theoretical purity.

Similarly, when beginning to design an API, be sure to understand and document the user need that the API aims to address.

Like all principles, this is not absolute. Ease of authoring affects how content reaches users. User agents have to prioritize finite engineering resources, which affects how features reach authors. Specification writers also have finite resources, and theoretical concerns reflect underlying needs of all of these groups.

See also:

- [The web should not cause harm to society](#)
- [The Internet is for End Users](#)

§ 1.2. It should be safe to visit a web page

Hyperlinks, links from one page to another, are one of the foundations of the Web.

Following a link, or visiting a web page, should be safe: users doing this should not have to fear for the security of their computer or for essential aspects of their privacy. (But it's not completely safe, in the sense that users may face consequences if their use of the Web is harming others.) Furthermore, users should understand that it is safe (and how it isn't) so they can make informed decisions between use of the Web versus other technologies.

Saying "essential aspects" here skips over quite a bit of detail. The Web today is far from being perfectly private. One aspect of privacy problems is when reality doesn't meet expectations. For example, a person walking down the street generally expects to be recognized by their friends, but (depending on the country) may not expect that they walked down that street at that time to be recorded in a permanent government database. Online, people might have less understanding of what is or isn't possible, and their

Abstract

Status of This Memo

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

Copyright Notice

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

1. Introduction
2. Who Are "End Users"?
3. Why the IETF Should Prioritize End Users
4. How the IETF Can Prioritize End Users
 - 4.1. Engaging the Internet Community
 - 4.2. Creating User-Focused Systems
 - 4.3. Identifying Negative End-User Impact
 - 4.4. Handling Conflicting End-User Needs
 - 4.5. Deprioritizing Internal Needs
5. IANA Considerations
6. Security Considerations
7. Informative References

IAB Members at the Time of Approval

Acknowledgements

W3C priority of constituencies

End users > Authors > Implementors > Spec editors > Theoretical purity

Size of each constituency

End users > Authors > Implementors > Spec editors > Theoretical purity

Billions Millions Thousands Tens

Cost of 1 hour of spec work

End users > **Authors** > **Implementors** > **Spec editors** > **Theoretical purity**
1,000,000,000 h *1,000,000 h* *100 h* *1 h*

Apache Software Foundation mantra

End users > Authors > Implementors > Spec editors > Theoretical purity

Apache Software Foundation mantra

End users > Authors > Implementors > Spec editors > **Theoretical purity**

Apache Software Foundation mantra

End users > Theoretical purity

Apache Software Foundation mantra

People > Code

What would an open source
priority of constituencies look like?

Open source priority of constituencies

People

Contributors

End users

App devs

Maintainers

*(Individuals &
corporations)*

Cloud infra

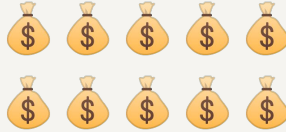
Open source priority of constituencies

People > End users > App devs > Cloud infra > Contributors > Maintainers > Theor. pur.

But is W3C's priority of constituencies a silver bullet?

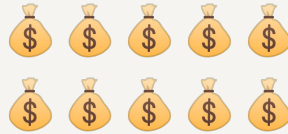
Economic situation

End users > Authors > Implementors > Spec editors > Theoretical purity



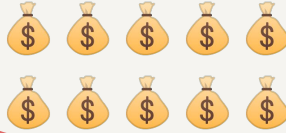
Economic situation

End users > Authors > Implementors > Spec editors > Theoretical purity



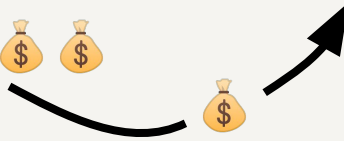
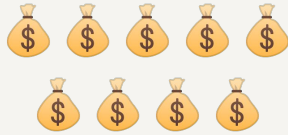
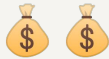
Economic situation

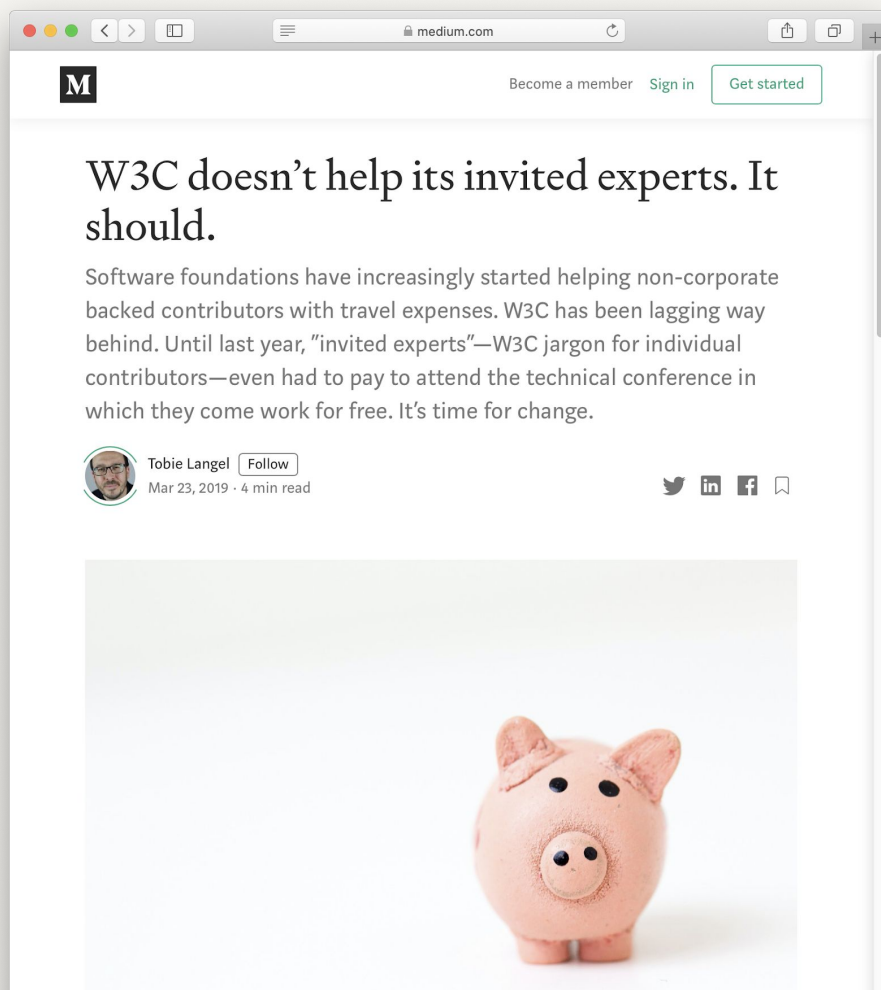
End users > Authors > **Implementors** > Spec editors > Theoretical purity



Economic situation

End users > Authors > Implementors > Spec editors > Theoretical purity





Let's apply this to open source.

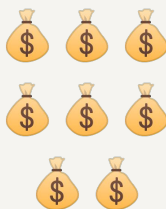
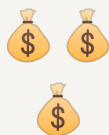
The canary in the coal mine

Open source priority of constituencies

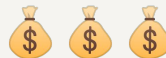
People > End users > App devs > Cloud infra > Contributors > Maintainers > Theor. pur.

Open source priority of constituencies

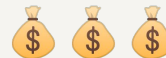
People > End users > App devs > Cloud infra > Contributors > Maintainers > Theor. pur.



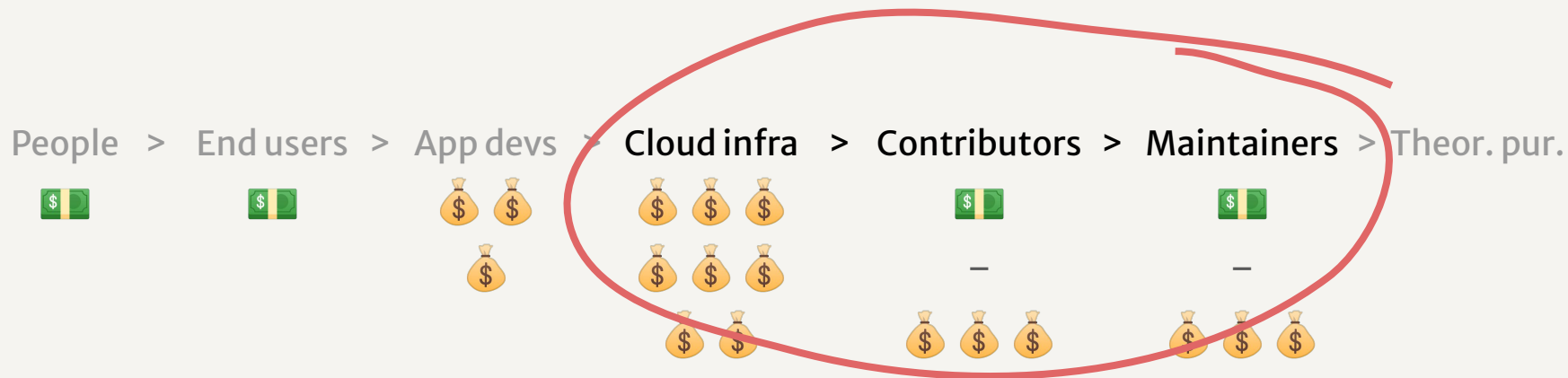
—



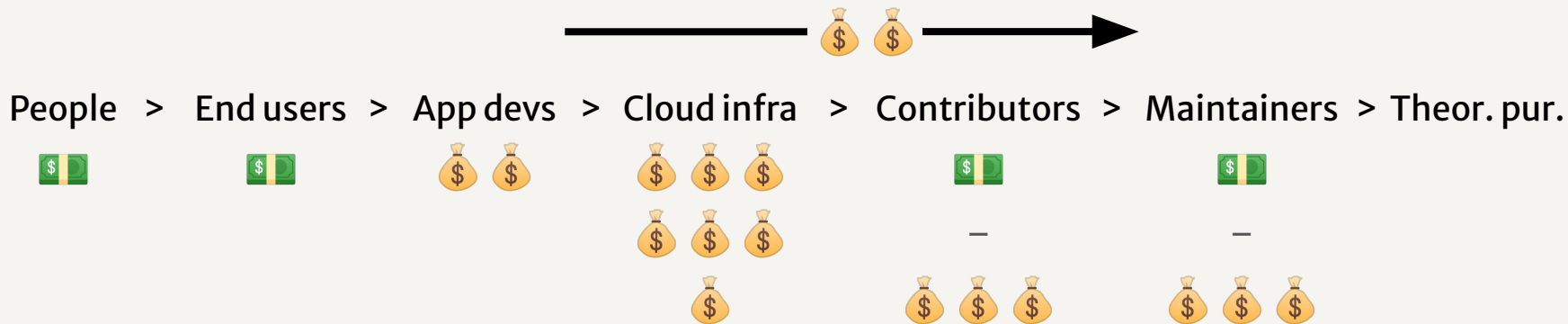
—



Open source priority of constituencies



Open source priority of constituencies



Benefits of a priority of constituencies

- Keeps focus on people & downstream impact.
- Maximizes benefits to the commons by upstreaming work.
- Surfaces discrepancies between economic situation and work expectations.