SCARF

Lessons Learned from Analyzing 1 **Trillion** Open Source Downloads

Percona Live 2026

Avi Press

Introduction

About me

- Founder & CEO of **Scarf**
- Based in Oakland, CA
- OSS maintainer, entrepreneur
- I really like functional programming (and promise not to mention it again)

About Scarf

- Founded in 2019
- Download metrics with artifact registry gateway and/or package telemetry
  - (We also do documentation and web analytics, out of scope for this talk)
- Works with both commercial and non-commercial open source projects
- Promote OSS sustainability and businesses with responsible, anonymized analytics

Talk Outline

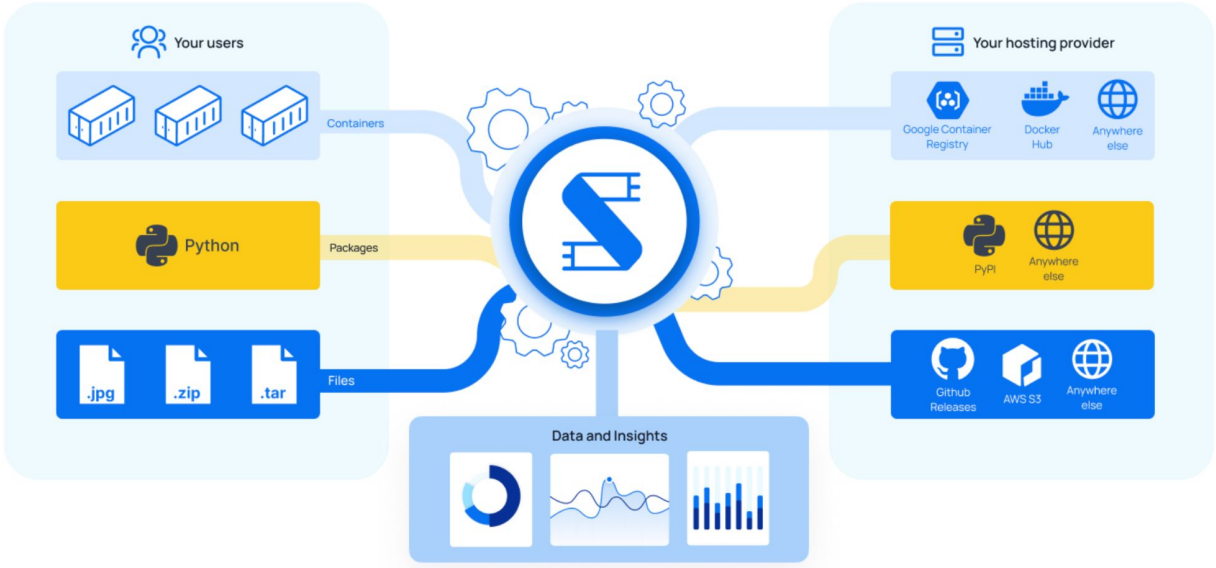
1. The data and how it was collected
2. Trends from the data
3. What we can learn and why it matters

About The Data

- 100k packages, primarily comprising:
  - Java Packages
  - Docker containers
  - Helm Charts
  - Binaries, tarballs, other files
  - npm packages
  - Python packages
  - Haskell packages
  - TF modules
- Package creators include:
  - Commercial and non-commercial OSS
  - Single vendor and multi-vendor
  - Significant OSS foundation-held OSS: Apache Software Foundation, Linux Foundation, CNCF, and others.
- 2.2 T total downloads
- 350M anonymized origin IDs (users*).
- IP address metadata from multiple sources, including WHOIS, IP Registry, Clearbit, 6sense.

How the data is collected

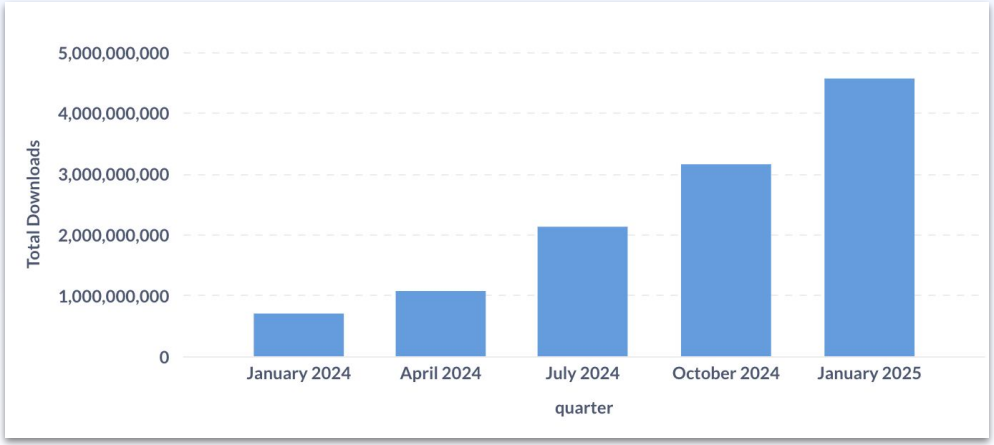
Scarf Gateway | Docker/OCI Containers | File, URLs, API Endpoints | Python Packages



How the data is collected

- Registry-level insights with Scarf Gateway
- PostInstall telemetry (eg, scarf-js)
- Registry integrations
- Metadata is processed, IP metadata is collected
- Data is anonymized, PII removed

About The Data: Total Volume



Total events / quarter

What does a “user” mean?

Identifying functions

```
endpoint_id(request)= hash(request.ip_address)
```

```
origin_id(request)= hash(request.ip_address,
request.user_agent, …)
```

Both origins and endpoints will overcount and undercount in different ways.

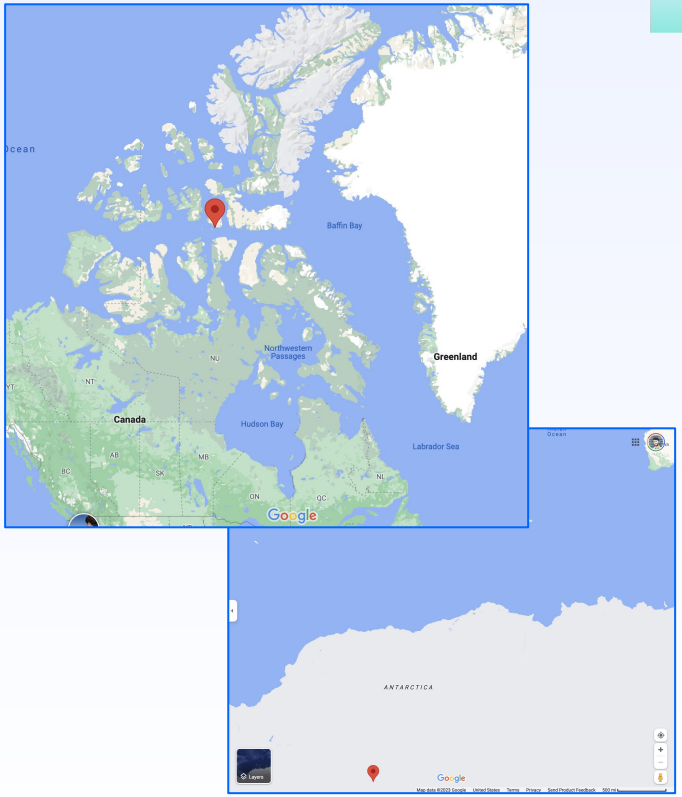
- 1000s of users behind the same IP
- 1 user on multiple IPs
- Distinct user agents may be different programs/machines/people

What can an artifact registry actually see?

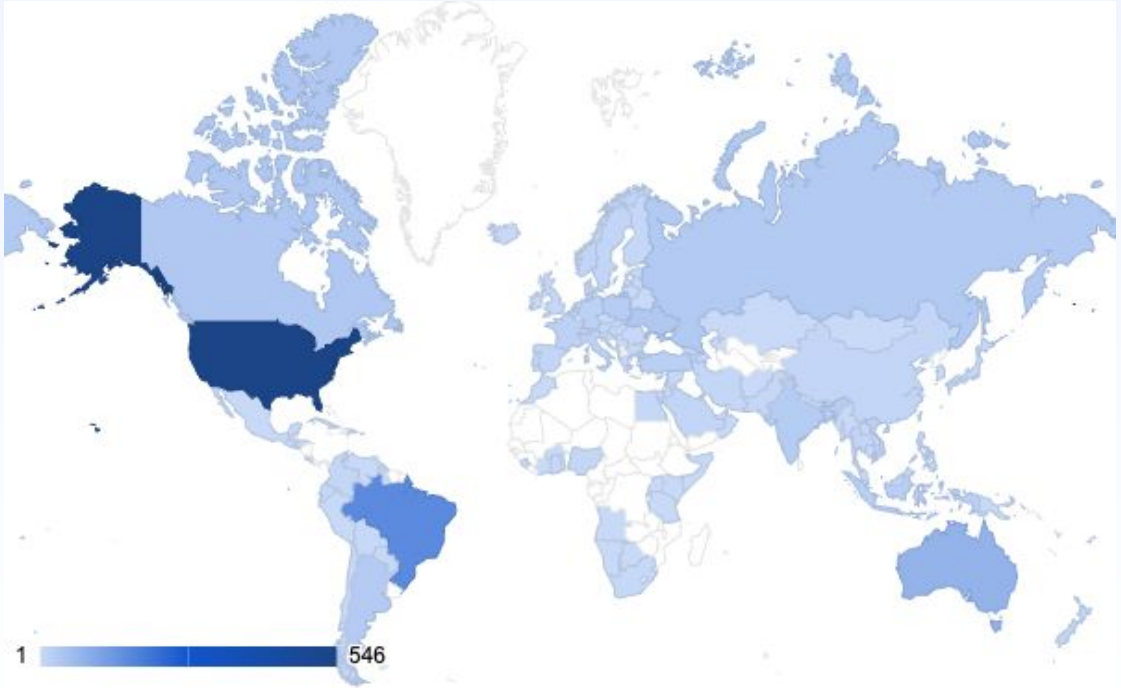
- What is being downloaded
- When it was downloaded
- User agent
- Other HTTP headers
- An IP address

Open source is used in every country, even some remote places!

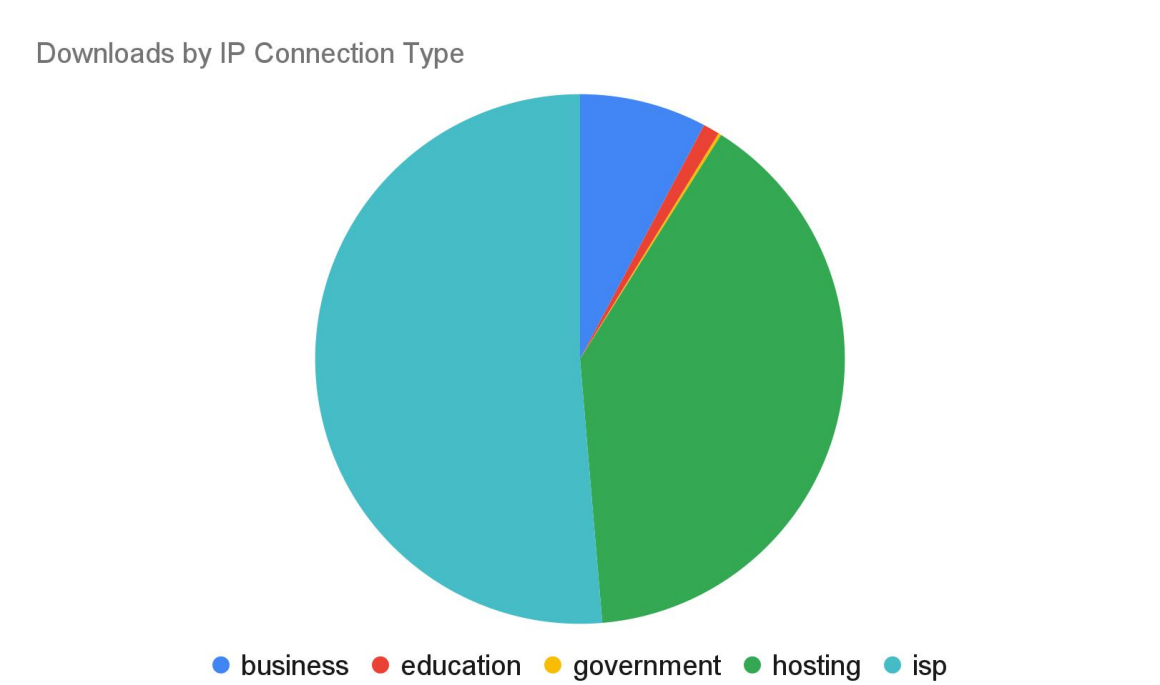
- As far north as Resolute, Canada
  - Closely followed by Nuussuaq, Greenland.
- As far south as deep Antarctica



Governments around the world use open source



What kinds of downloads are we seeing?



2M

Corporate-associated unique endpoint_id domains. There is a wealth of data here to support commercial open source businesses!

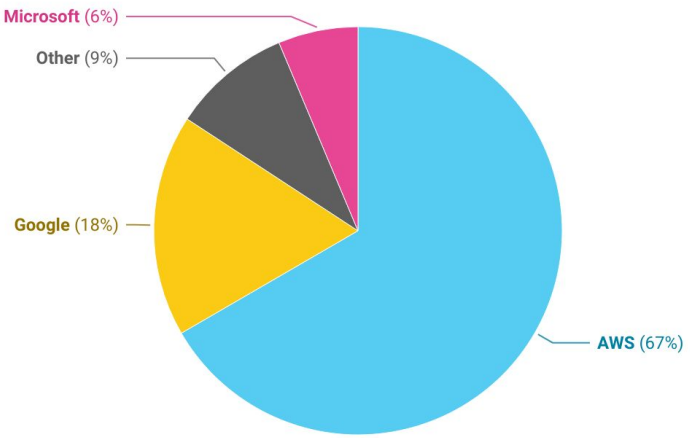
96%

of the Fortune 500 seen downloading OSS packages

Which public clouds?

Unique Events by Public Cloud

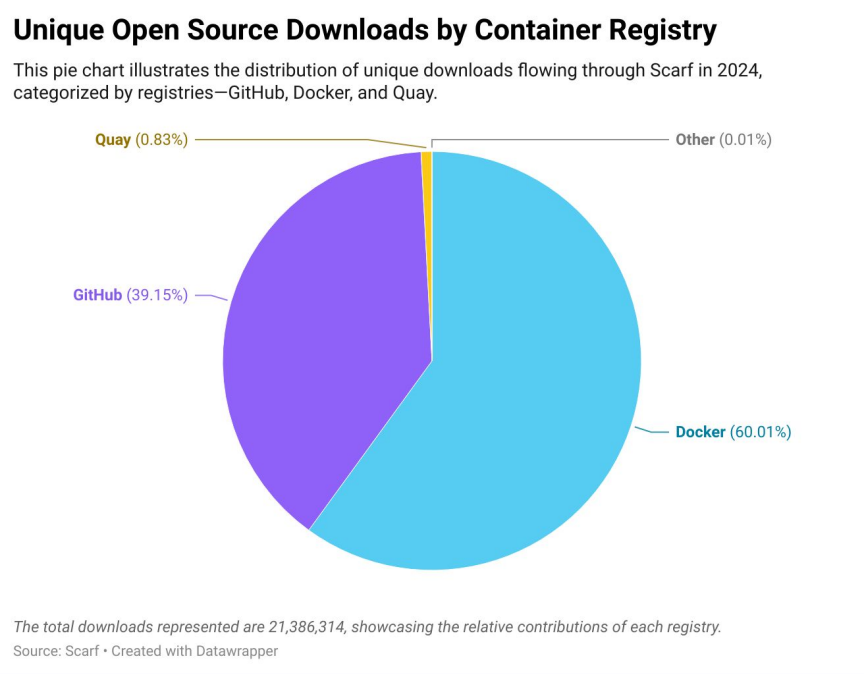
This pie chart compares the distribution of total open source events flowing through Scarf in 2024 across the three largest hosting providers.



These "big three" stand out among the 9,789 hosting providers globally identified by Scarf, highlighting their significant role in open source distribution.

Source: Scarf • Created with Datawrapper

Container registry market share (from Scarf)



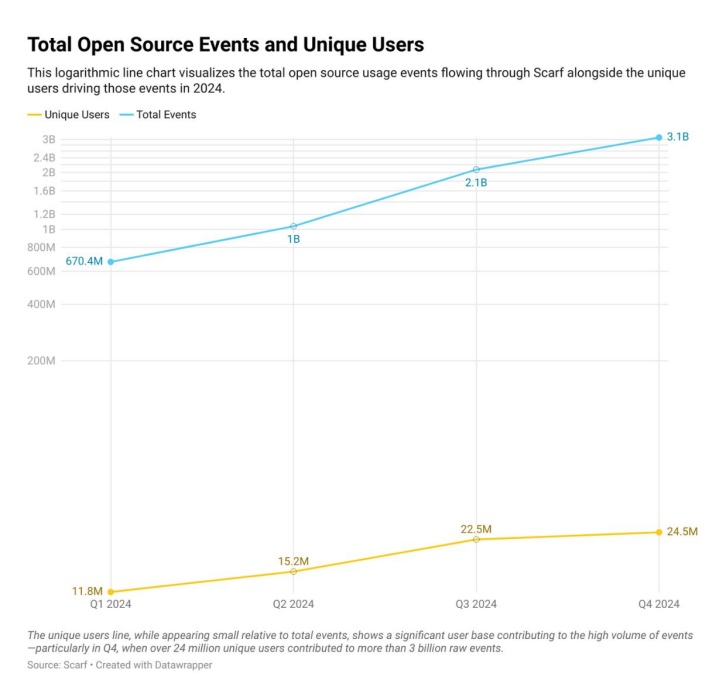
0.1%

of IPs downloading packages were from a recognized VPN

94.8%

Portion downloads were from the 1% of projects

Downloads vs unique users looks very different



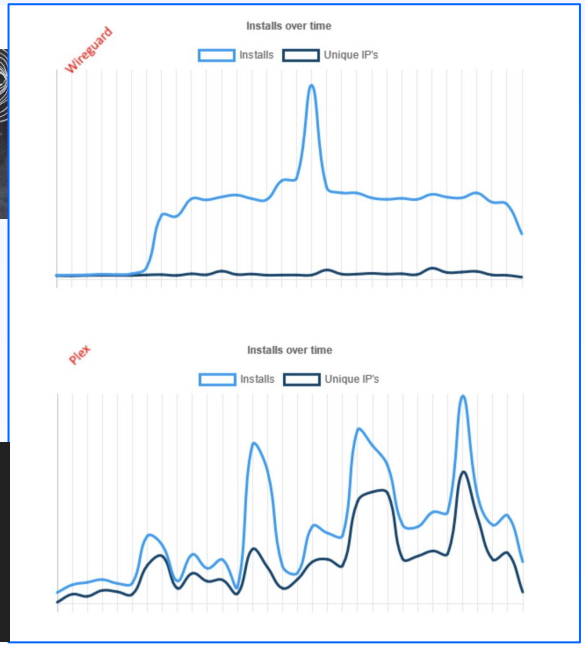
Downloads vs unique users looks very different

Average ratio of downloads to unique users: ~100:1

Best packages see closer to ~15:1

Many surges in downloads are not real user growth

“Almost half of our pulls through Scarf can be attributed to 20 users with misconfigured or overly aggressive deployment/update services. As you can see this doesn’t apply across the board, with most images closely tracking pulls with unique users.”



Many surges in downloads are not real user growth

How much time and resources have been wasted by not understanding this?

A lot of your downloads are from automated systems

- Container agents that continually pull for updates
- CI/CD
- Artifact mirrors
- For tarballs and things that people download in a browser, web crawlers are another factor.

Top clients (red denotes repetitive downloaders)

- GoHttpClient
- Docker
- containerd
- Unknown
- RenovateBot
- cri-o
- skopeo
- containers
- go-containerregistry
- Python aiohttp
- curl
- Harbor
- Helm
- Diun
- UptimeRobot
- Renovate Bot
- Python Requests
- urlgrabber
- GoogleStackdriverMonitoring-UptimeChecks

More clients should set rich headers / user agents

Pip does an exceptionally great job


```
pip/23.2.1 {
    "ci": true,
    "cpu": "x86_64",
    "distro": {
        "id": "jammy",
        "libc": {
            "lib": "glibc",
            "version": "2.35"
        },
        "name": "Ubuntu",
        "version": "22.04"
    },
    "implementation": {
        "name": "CPython",
        "version": "3.10.13"
    },
    "installer": {
        "name": "pip",
        "version": "23.2.1"
    },
    "openssl_version": "OpenSSL 3.0.2 15 Mar 2022",
    "python": "3.10.13",
    "rustc_version": "1.72.0",
    "setuptools_version": "65.5.0",
    "system": {
        "name": "Linux",
        "release": "6.2.0-1011-azure"
```


Why it's great:

- CI Flag!
- System information
- Build information
- Dynamic dependency information
- Human readable
- Machine readable

More clients should set rich headers / user agents

Pip does an exceptionally great job

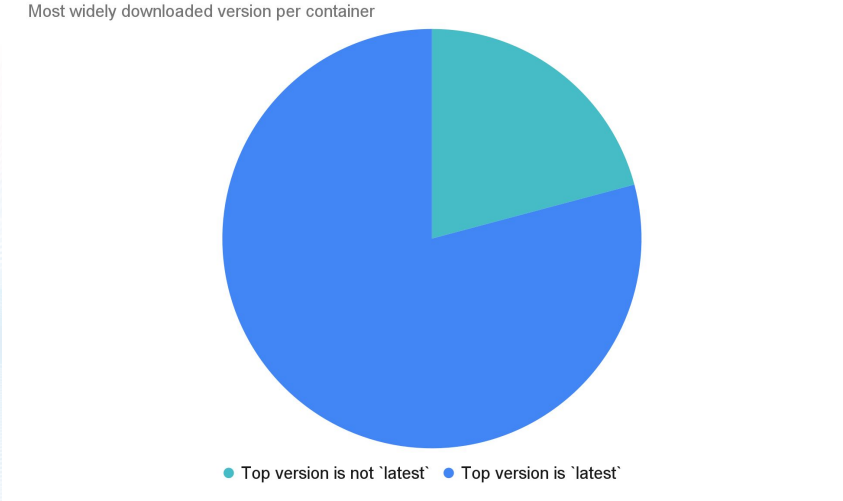

```
pip/23.2.1 {
    "ci": true,
    "cpu": "x86_64",
    "distro": {
        "id": "jammy",
        "libc": {
            "lib": "glibc",
            "version": "2.35"
        },
        "name": "Ubuntu",
        "version": "22.04"
    },
    "implementation": {
        "name": "CPython",
        "version": "3.10.13"
    },
    "installer": {
        "name": "pip",
        "version": "23.2.1"
    },
    "openssl_version": "OpenSSL 3.0.2 15 Mar 2022",
    "python": "3.10.13",
    "rustc_version": "1.72.0",
    "setuptools_version": "65.5.0",
    "system": {
        "name": "Linux",
        "release": "6.2.0-1011-azure"
```


Others examples

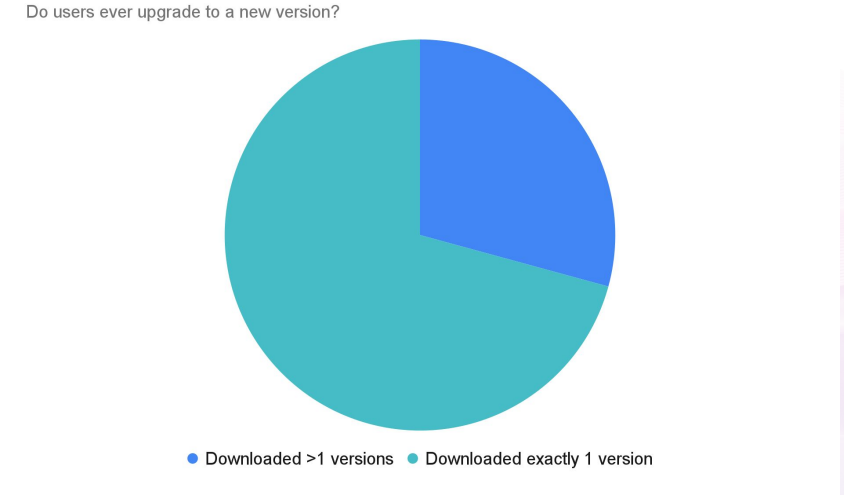
- Good:
  - `docker/20.10.7 go/go1.13.15 git-commit/b0f5bc3 kernel/5.19.2-x86_64-linode156 os/linux arch/amd64 UpstreamClient(Go-http-client/1.1`
  - `Homebrew/4.1.14 (Macintosh; arm64 Mac OS X 13.4) curl/7.88.1`
- Less good:
  - `GoHttpClient`

People do not upgrade versions like you might expect

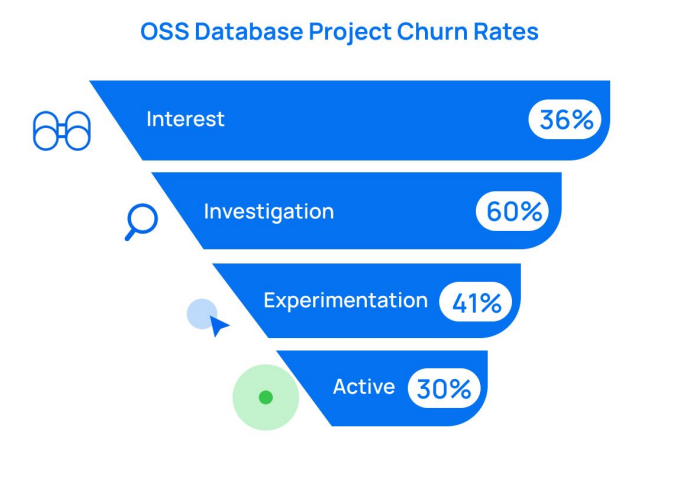
`latest` is the most downloaded version for 80% of containers available on Scarf.

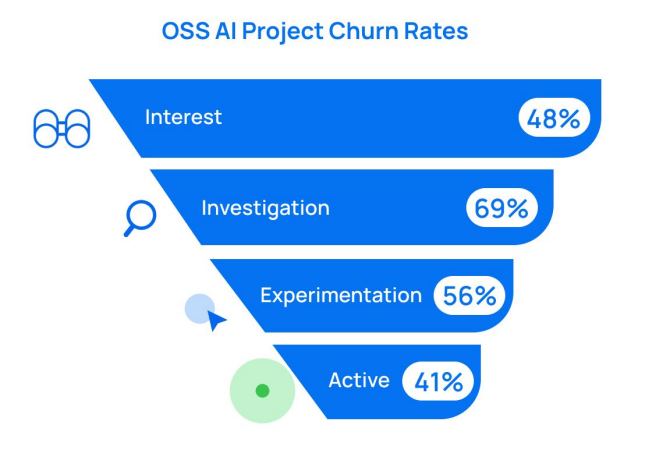


Most users will never download a second version, even if they continue using the software



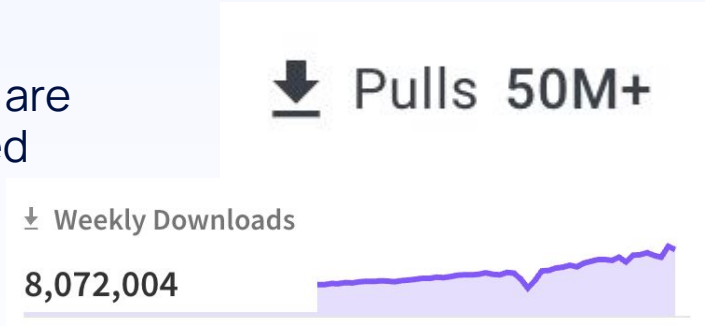
A tale of two communities: AI vs Databases





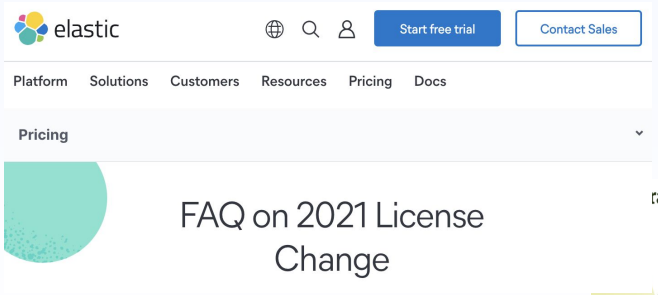
Why does any of this matter?

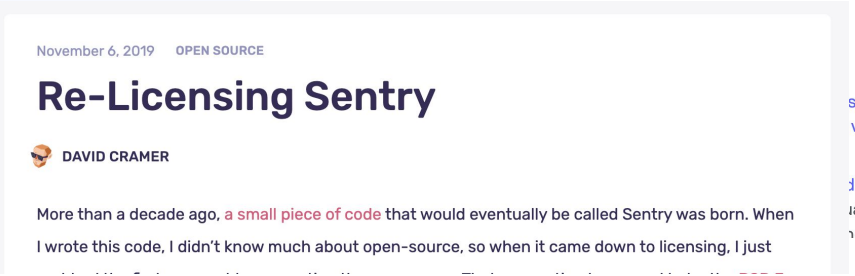
- Open source work can be done more effectively with data.
- Changing attitudes on data collection in open source can be good for the whole ecosystem.
- The current status quo for usage metrics are misleading the entire industry if not looked at more critically.
- The data *already exists* we just aren't leveraging it.

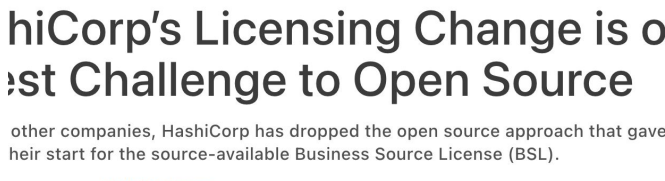


Why does any of this matter?

Want to see fewer companies ditch pure Open Source licenses? We need to help businesses commercialize OSS more effectively, and analytics will be essential to that end.







Takeaways

- Open source is really everywhere. Your work in OSS has a huge impact on the world!
  - Big businesses
  - Governments
  - Educational institutions
- Tracking the usage of your OSS can be critical to building a thriving business around it.
- Total download counts alone can be highly misleading. Always pair with some notion of uniqueness if possible.
  - Huge outliers in traffic
  - Lots of bots
  - Redownloads
- Please put more (machine readable) information into your user agents
- Maintainers: Keep realistic upgrade habits of users in mind.
- Usage data can help the entire OSS ecosystem. Let's embrace it and work to do it responsibly*!

(*a topic for a whole other talk)

SCARF

Thank you!

Email: avi@scarf.sh
Twitter: @avi_press
GitHub: @aviaviavi
Linkedin: (Avi Press)

Scarf

Avi