The past and the future of the PostgreSQL community

Tomas Vondra, EDB, <tomas@vondra.me>

Swiss PGDay 2024, June 27-28

Who here uses Postgres just because it's cheaper, and who actually cares about the "open source" idea?

What I'm going to talk about ...

- history
  - Where did the community come from?
  - How did that affect the structure of the community?
- current state
  - State of the community.
  - The ~~arcane~~ traditional development process.
- future
  - Where will we (maybe) go?
  - How will we adjust the development process?

history

the distant past

- 1986 - POSTGRES project starts at Berkeley
  - evolution of earlier research projects, various improvements over the years
- 1995 - released as Postgres95 as open source
  - a lot of improvements on top of POSTGRES, reworks, portability, stability, ...
- 1996 - maybe Postgres95 was not such a great idea?
  - rename to PostgreSQL, CVS history starts at 1996/07
  - this is where the core team comes from - the OGs of PostgreSQL
  - they did everything, they were the only people willing to do so
  - also beginning of the original development process

not so distant past

2004 - buildfarm started (https://buildfarm.postgresql.org)

2008 - maybe we should stop doing waterfall development

- very unpredictable releases (not even a waterfall, really)
- invention of commit fests (monthly cycles, wiki)

2009 - let's organize commitfests better (not just a wiki page)

- https://commitfest.postgresql.org/ [old]

2010 - let's switch from CVS to git

- https://lwn.net/Articles/409635/

1: emergent structure of the community

- things evolved a certain way, which determines the structure
  - a "green field" design might look differently, but well ...
- things are not set in stone
  - core team does not want to do everything - CoC, committers, ...
  - many independent bits of the community (PUGs, various related projects, ...)
- it's not about "just core"
  - relationship of multiple orgs, many other parts of the community
- people are aware of the challenges

2: emergent development process

- originally not "arcane", this is how distributed development worked
    - mailing lists were the common way to share patches (e.g. kernel)
- we did actually do various improvements over the years
    - commitfests were a huge step forward
    - regular release cycle was another major improvement
    - the commitfest app (CFA) helped a lot too
- mailing lists are not the norm anymore, so seems a bit arcane
    - people no longer know how to do this, may need some adjustments
    - do the proposed changes address the actual bottlenecks?

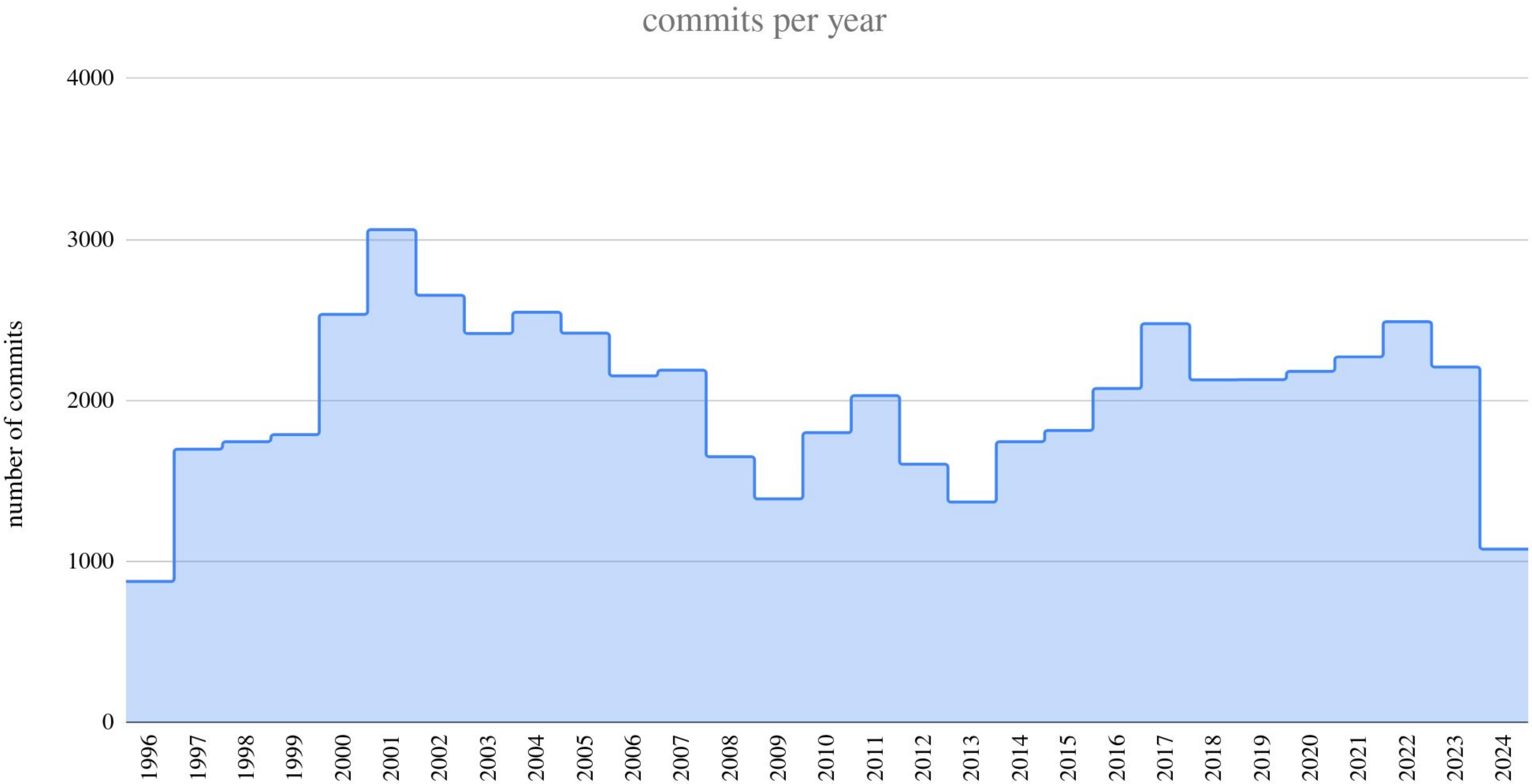
3: our tools precede a lot of modern stuff

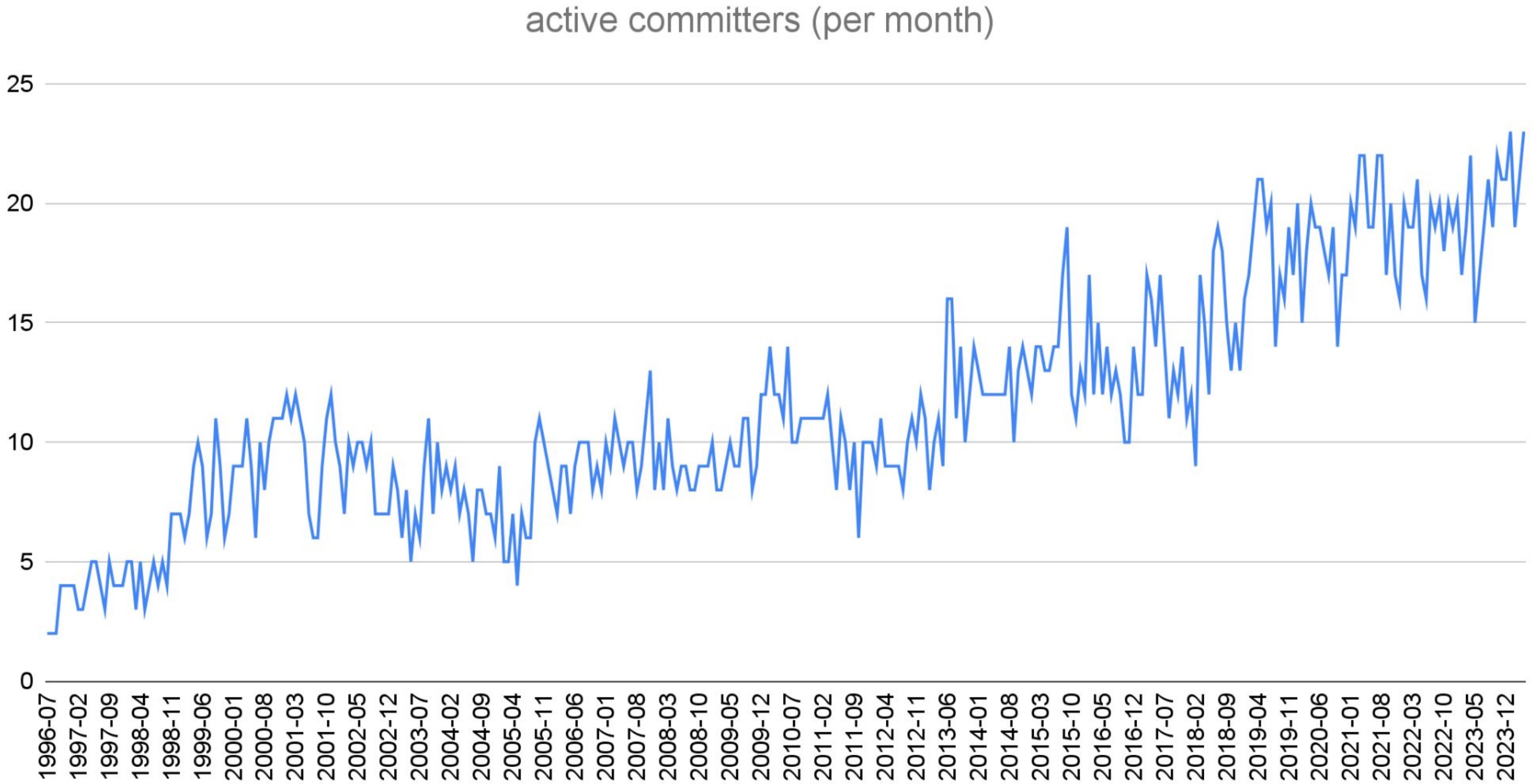
- e.g. github started in 2010
- also a lot of stuff (mostly) died meanwhile
  - think about sourceforge, pgFoundry (GForge), ...
- there still is not a 100% replacement for some stuff
  - buildfarm (our custom CI) runs on many exotic platforms
  - still nothing comparable
- reasons to prefer running our stuff on our infrastructure
  - e.g. legal reasons, commercial interests, other risks, ...

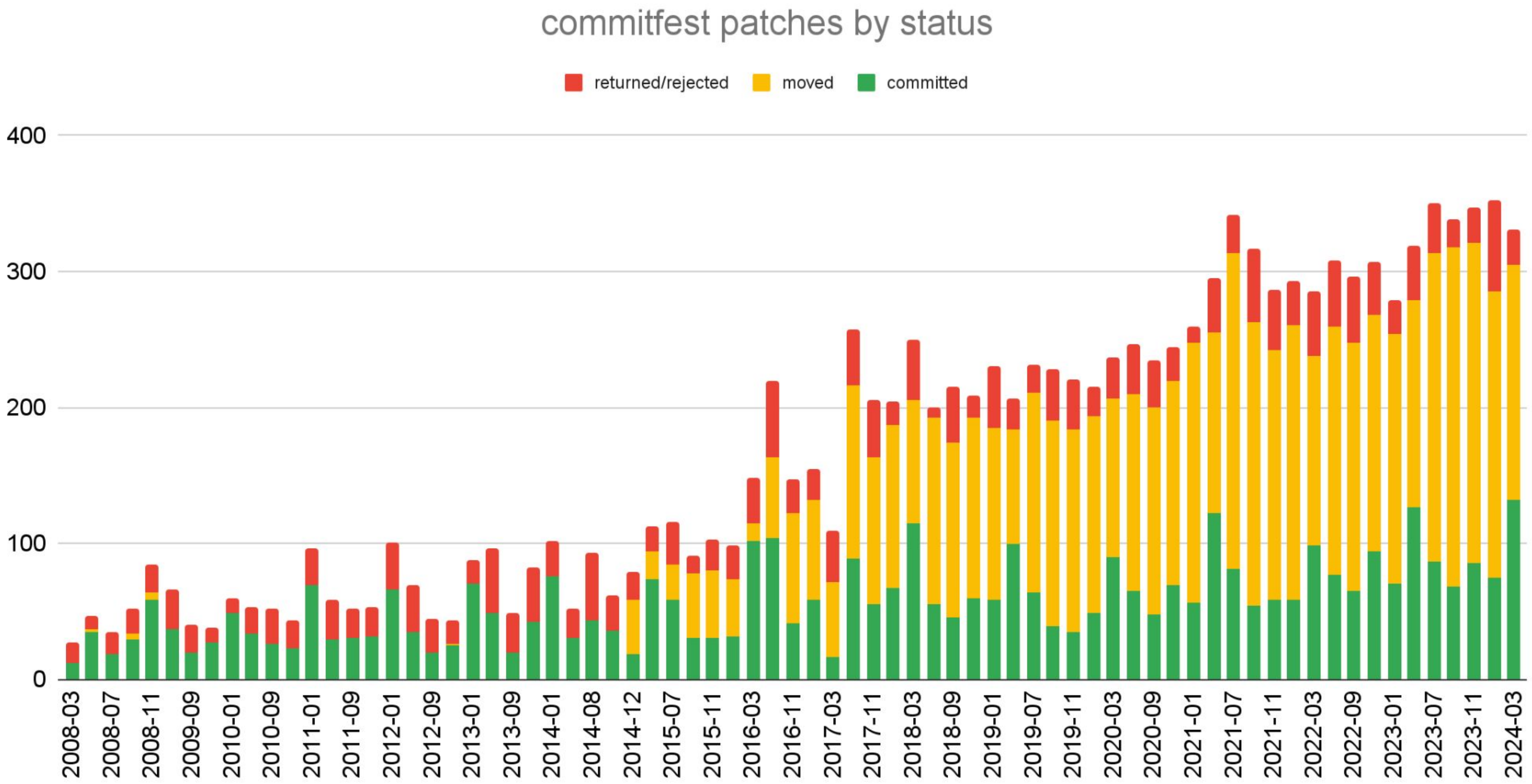
4: what about companies?

- I didn't actually mention any companies backing this, right?
  - no company controls the project
  - many companies contribute
- There are other projects like this (kernel, Libre Office, ...) of course.
- But many projects are dominated by a single company.
  - That has risks, of course. Also, makes it harder to collaborate in open source manner.
- PostgreSQL is super-friendly to forks (and we have quite a few).
  - A bit ironic that we have very few issues with forks. It's a strength.

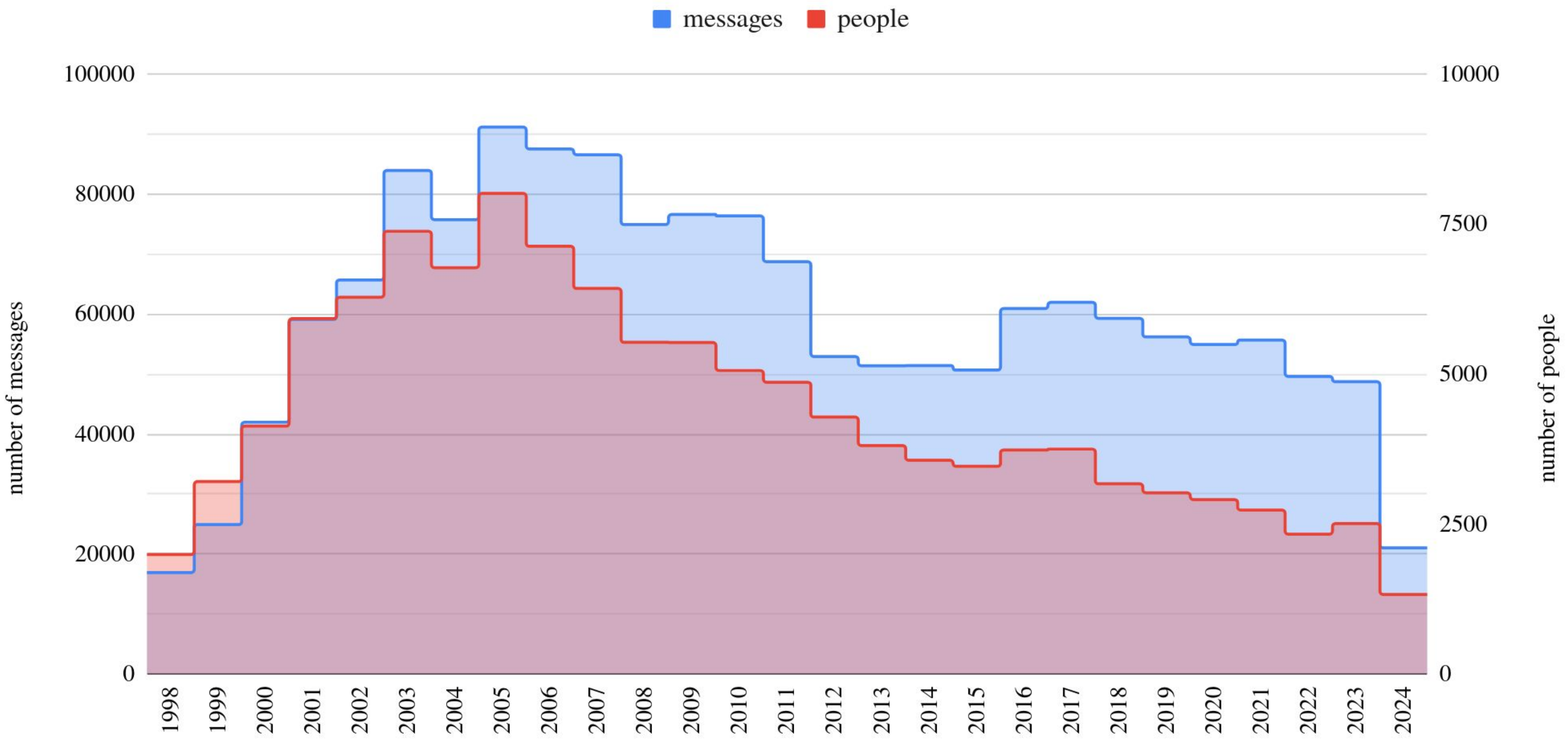
present

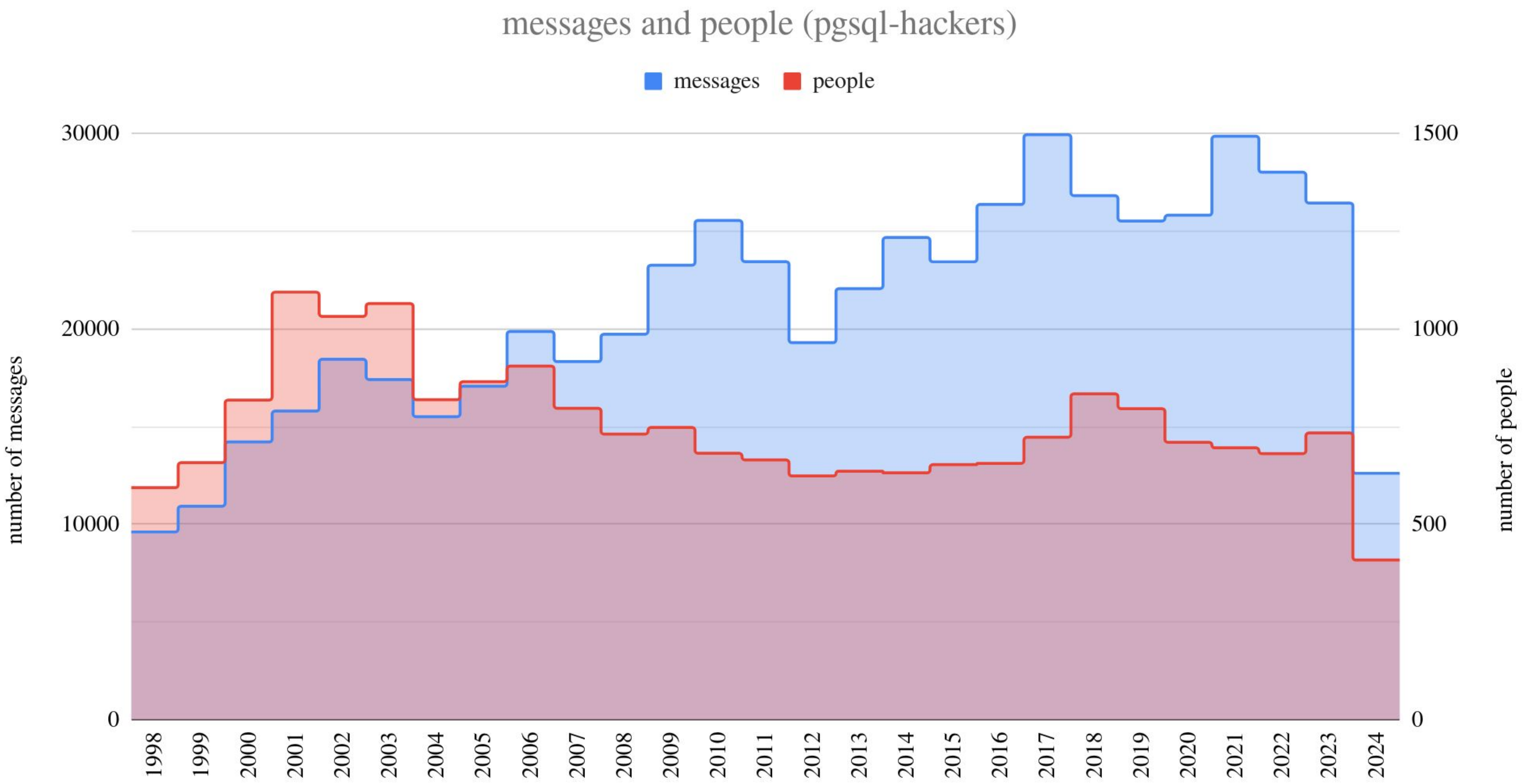

commits per year
number of commits
4000
3000
2000
1000
0
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024


active committers (per month)
25
20
15
10
5
0
1996-07
1997-02
1997-09
1998-04
1998-11
1999-06
2000-01
2000-08
2001-03
2001-10
2002-05
2002-12
2003-07
2004-02
2004-09
2005-04
2005-11
2006-06
2007-01
2007-08
2008-03
2008-10
2009-05
2009-12
2010-07
2011-02
2011-09
2012-04
2012-11
2013-06
2014-01
2014-08
2015-03
2015-10
2016-05
2016-12
2017-07
2018-02
2018-09
2019-04
2019-11
2020-06
2021-01
2021-08
2022-03
2022-10
2023-05
2023-12

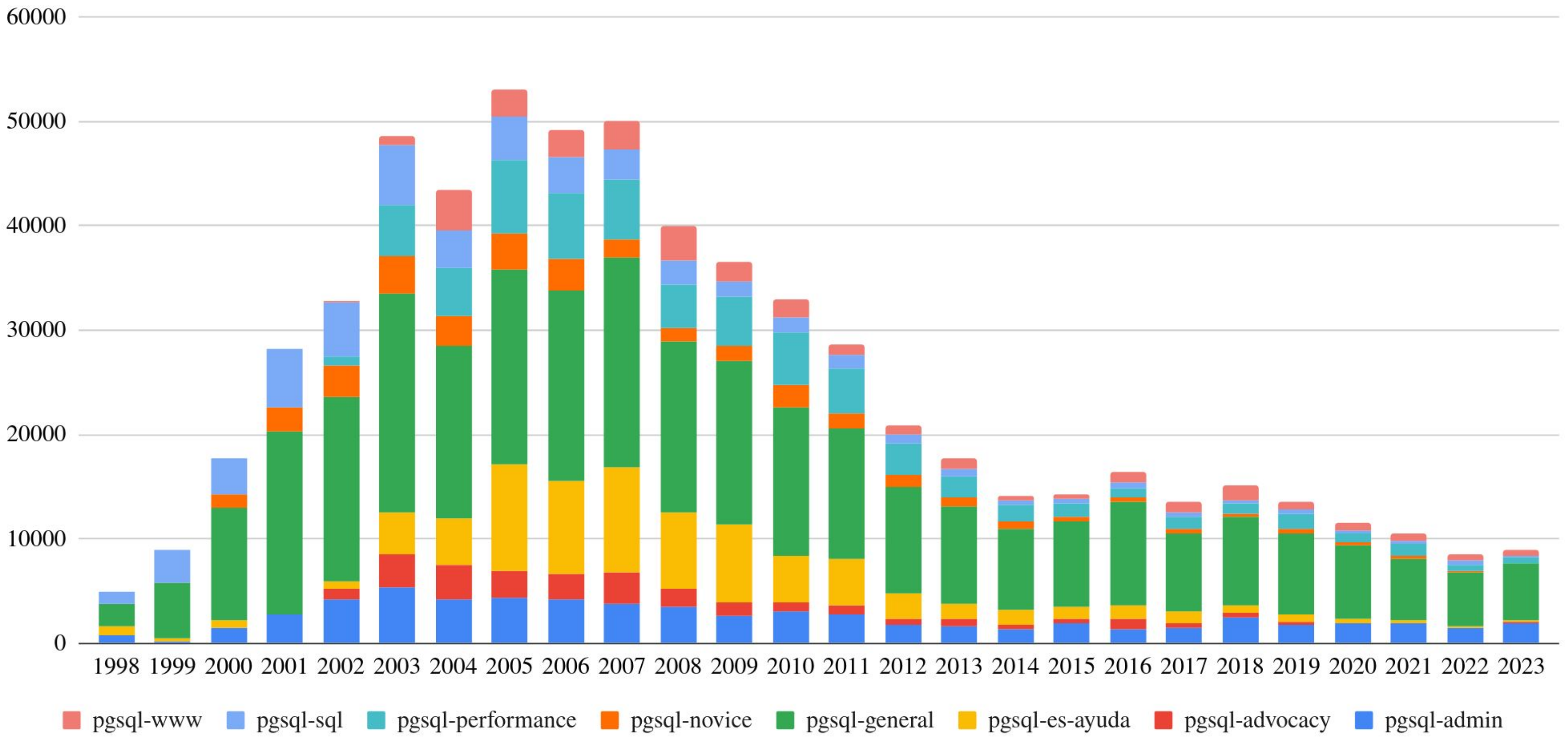

commitfest patches by status
🟥 returned/rejected
🟨 moved
🟩 committed
400
300
200
100
0
2008-03
2008-07
2008-11
2009-09
2010-01
2010-09
2011-01
2011-09
2012-01
2012-09
2013-01
2013-09
2014-01
2014-08
2014-12
2015-07
2015-11
2016-03
2016-11
2017-03
2017-11
2018-03
2018-09
2019-01
2019-07
2019-11
2020-03
2020-09
2021-01
2021-07
2021-11
2022-03
2022-09
2023-01
2023-07
2023-11
2024-03

messages and people (all mailing lists)




messages and people (pgsql-hackers)
messages
people
number of messages
number of people
30000
20000
10000
0
1500
1000
500
0
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024

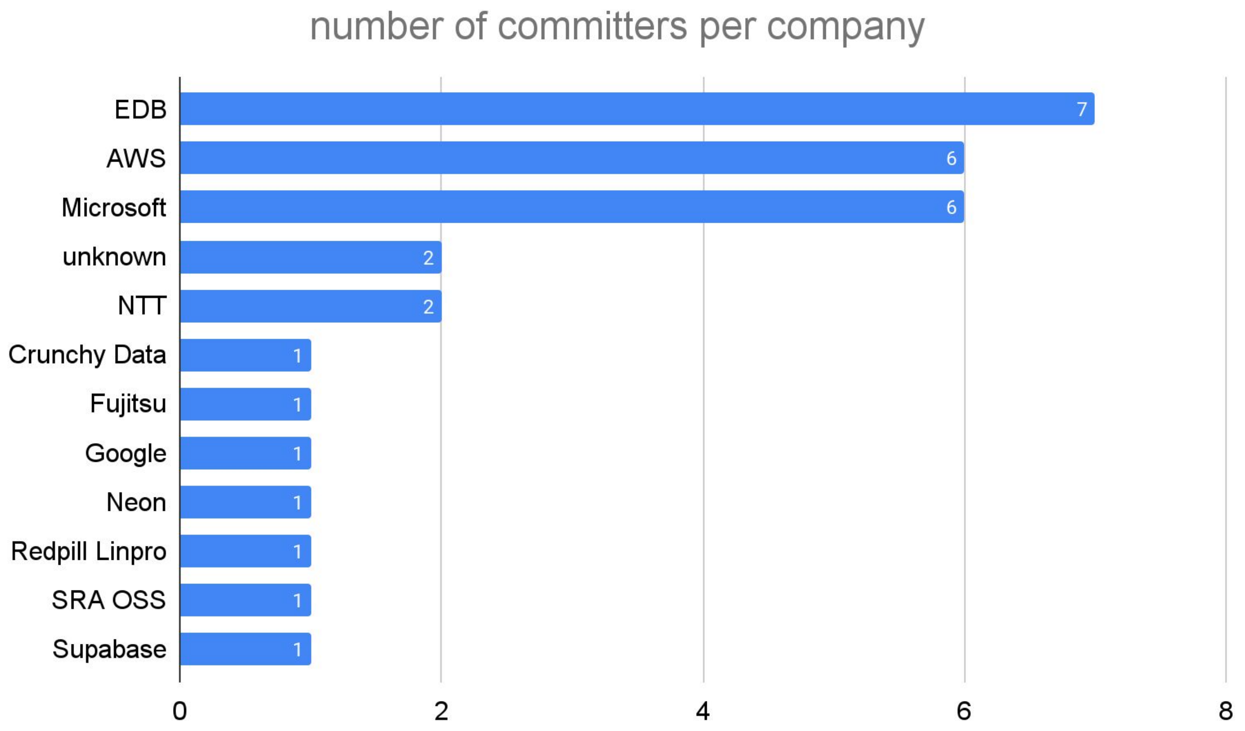
messages per year / less active mailing lists



Takeaways?

- development activity is growing
  - more committers, more messages on pgsql-hackers, ...
  - number of committed patches stagnates
  - review bottleneck? increased complexity? stricter reviews?
  - development process may need improvements
- the activity on "user" lists declined a lot
  - likely moved elsewhere (SO, slack, discord, ...)
  - we may be losing an important source of feedback

A bit about companies ...



future

community structure

- the community structure will continue to evolve
  - core will delegate more stuff to other people
  - likely in direction to less "self selection", also a matter of trust
- not everything has to be driven by core
  - community = gathering of individuals with agency
  - don't wait for core to do something, do it yourself + ask for help
- users clearly moved elsewhere
  - not a bad thing, but we need to adapt (not pretend nothing changed)

development process

- incremental changes / evolution [discussion]
  - foundation likely to remain the same (regular cycles, ...)
  - make it more accessible for "current generation" of devs
  - maybe some "federation" (accept PRs, curate/forward to hackers?)
- really need to address the bottleneck(s)
  - more committers? takes time, not sure if solution on it's own
  - "growing" contributors able to review complex stuff (mentoring)
  - make reviews easier (clear patch status)

growing contributors

- mentoring of future contributors
    - responsibility of current committers
    - requires effort and motivated developers
    - community-driven efforts [thread] (but you can reach out directly too)
- importance community-wide mentoring
    - companies usually have internal mentoring, mixed teams, ...
    - that doesn't help to grow committers in smaller companies
    - risk of domination, focus on helping people in smaller companies

EDB
Postgres for the AI Generation

Swiss PGDay 2024 Sponsor

Enter to Win

Take part in the EDB prize draw
to win a brand new book.

PostgreSQL 16
Administration
Cookbook

Solve real-world database administration challenges
with 180+ practical recipes and best practices

SCAN THE QR CODE
AND FILL IN THE FORM
TO ENTER

For more information about EDB visit www.enterprisedb.com

open source / free software

- unlikely to convince people to stop using managed services
  - convenience, cost, reduced expertise requirements, ...
  - but lets be very clear about the limitations and that "it's not Postgres"
- doesn't mean we can't educate people about the open source idea
  - developers often don't have experience with how our community works
  - users often just consume the product as is, don't realize it can be extended
  - ...

resources

Who are the contributors?

- no specific requirements for contributors (committers, ...)
- people take often very different paths
- maybe check https://postgresql.life/
  - interviews with community members
  - very different paths, very diverse areas of focus/interest

A bit about technology?

- didn't want to talk about individual features
- there's usually a talk about new stuff in each release
- but maybe look at the following two talks
- 2013 pgconf.eu keynote / Keith Alsheimer / EDB
  - https://momjian.us/main/writings/pgsql/other/KeynotePPEU_v9.pdf
  - 5-year prediction on slide 18, talk a bit about what we improved etc.
- 2023 pgconf.eu keynote by Simon Riggs
  - https://www.youtube.com/watch?v=8W-J36IxYv4
  - next 20 years, too early to review
  - definitely worth looking at

considering contributing?

Not sure how / where to start?

- Need an idea for your first patch?
- Already have an idea, but need advice / guidance / feedback?
- Not familiar with our development process?
- Something else?

Maybe talk to me ...

intro

- very different talk from what I usually do
  - usually - technical, this is much less so
  - also my first keynote, I hope not to ruin it
- a lot of this is an opinion / interpretation
  - my usual technical talks are backed by numbers/facts
  - I have many opinions, a lot of them are likely wrong
  - I'm speaking for myself, not for EDB or other PG contributors
- feel free to shout questions, I hate boring lectures
  - or approach me later and we can chat
  - I'm here to ask questions, not give answers.

agenda

- where does postgres come from
- how the community evolved
    - how the core team started, why it works the way it does
    - core team - committers - contributors - how will it evolve?
    - concerns about abrupt changes
    - if we started on a green field ... it'd look different
- some numbers on how the community evolved
    - number of committers, contributors, authors, commits, lines of code, ...
    - what about "users"?
    - maybe extract names from release notes?
    - statistics of pgsql-hackers

agenda II

- postgres vs. companies
  - no company "owns" postgres
  - a mix of companies, global/regional, small/large, ...
  - thankfully some of the large companies are joining and contributing
- postgres vs. forks
  - many forks, but vast majority being very friendly with the project
  - funny how well this works, compared to projects with licenses that make forking harder