Mageia, successes and lessons learned
6 years after forking

Anne Nicolas – Samuel Verschelde

A brief presentation of Mageia

How to define Magiea community and distribution

Essentials

- A GNU/Linux-based, Free Software operating system, easy for both users and contributors
- A community project, supported by a nonprofit organisation of elected contributors

Some figures

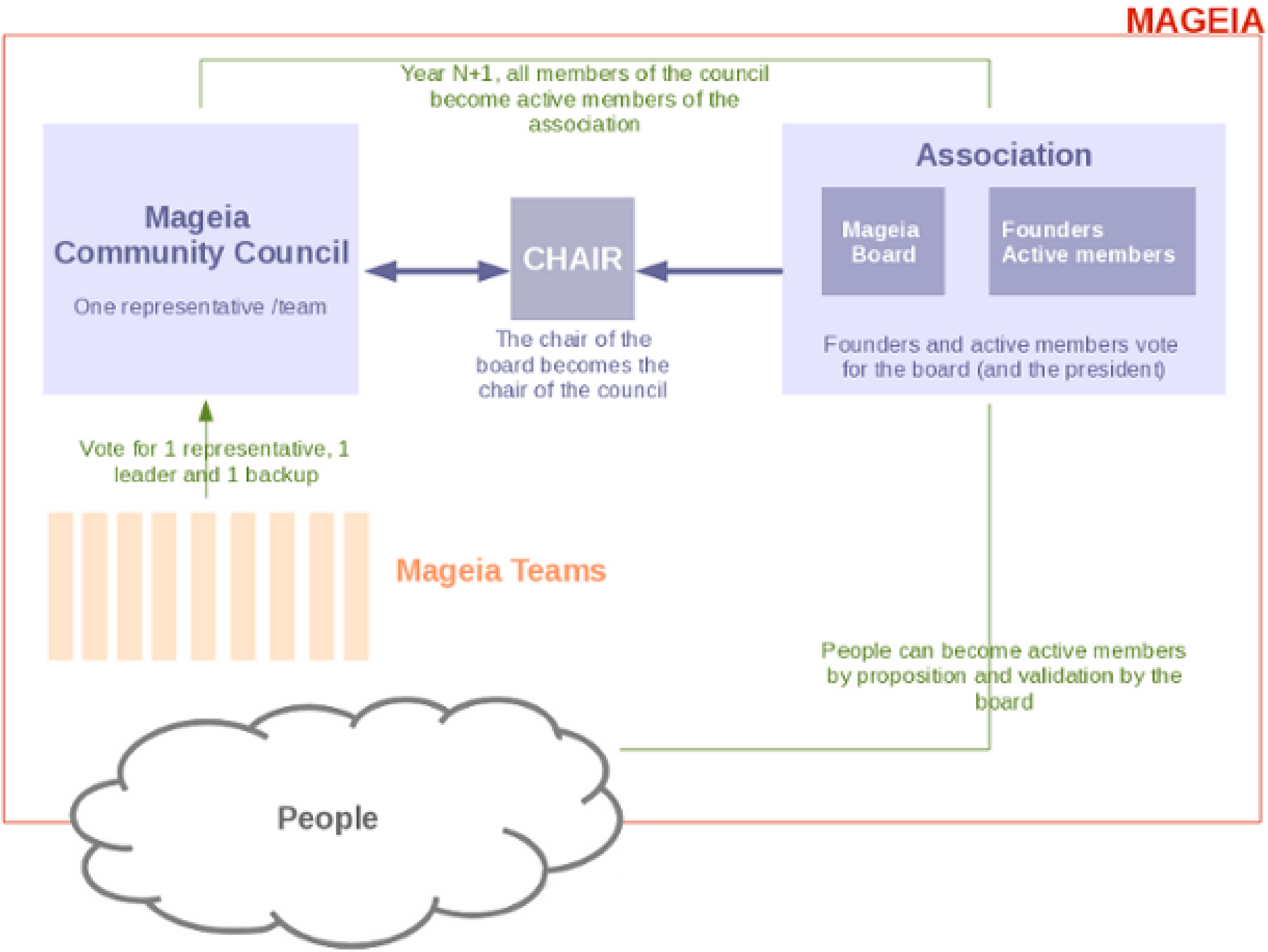
- 100 packagers accounts, 3 000 Bugzilla accounts, more than 20 000 bugs
- dev mailing-list (250), general dicussion mailing-list (300), forums (5 600 accounts, 66 000 posts)

Why this fork

What we wanted

- Give back the control to the community
- Setup all the project organization: governance, teams and collaboration
- Setup a technical and dedicated infrastructure (reliable, maintainable, scalable, replicable)
- Ship a first independant, stable and reliable version without major innovation, upgradable from Mandriva Linux
- Long term goals were not that defined except rescue more than 10 years of history

Our organization



MAGEIA
Year N+1, all members of the council become active members of the association
Mageia Community Council
One representative /team
CHAIR
The chair of the board becomes the chair of the council
Association
Mageia Board
Founders Active members
Founders and active members vote for the board (and the president)
Vote for 1 representative, 1 leader and 1 backup
Mageia Teams
People can become active members by proposition and validation by the board
People

The autocongratulation slide

What we achieved and we are proud of it

- 5 distributions have been released and 6th is on it way
- A real community of users **and** contributers
- No nasty issues for now on released versions
  no apocaliptic headline ever:)

OK... We may now be using the debian way of releasing : **« released when ready »**

From user to contributer

Lowering the entry barriers

Facts

- A healthy community project always needs to renew or grow its contributors base.
- Need to attract potential contributors and have them stay.

In Mageia

- Welcoming teams (even packagers).
- Joining teams is not just for elite. Anyone motivated and able to work in a team is welcome.
- Mentoring. Help them start in good conditions and learn our way of working together.

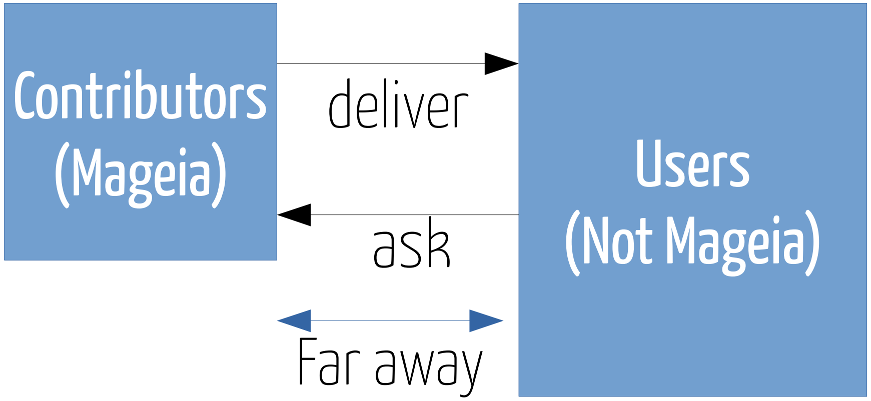
The barrier is usually higher in users heads than it is in reality.

- Belief that it requires extraordinary skills to contribute to such a big project.
- Some users have a habit of considering themselves as "outside" (a client mentality?).

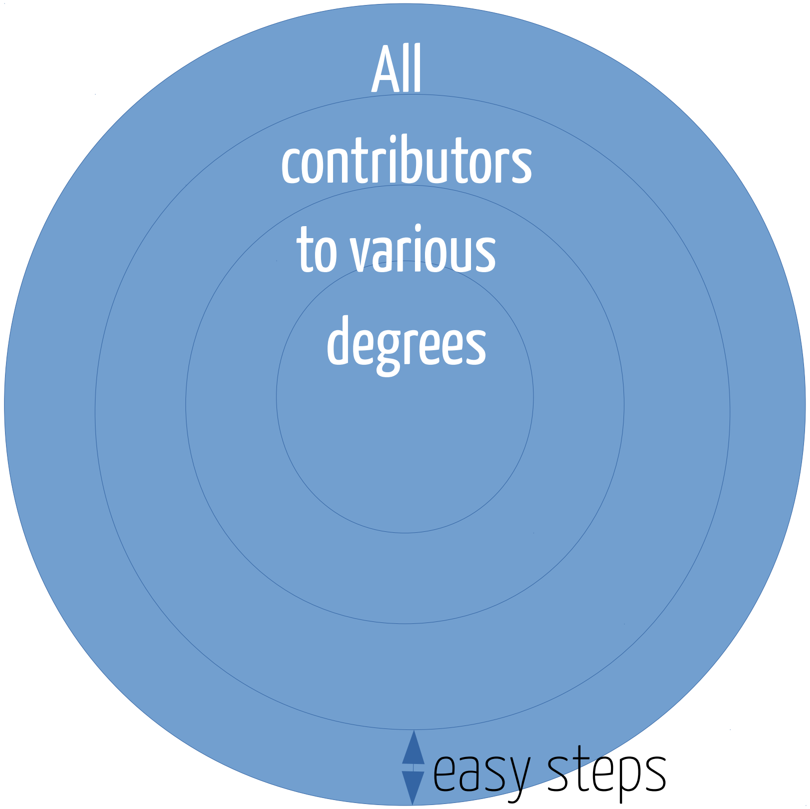
=> communicate and invite

Lowering the entry barriers

What many users think, by habit



How it is actually



Mageia as a community

A particular form of contribution: leadership

- Building a linux distribution: lots of different projects in one, need coordination.
- A need for leaders to show the way and act as catalysts.
- In mageia: elected team (co-)leaders and deputies. Almost all of them started as simple users.
- Also: anyone helping people to work together is doing unofficial leadership.

An example of unofficial leadership

- There's been an **annoying release-blocker** bug for months. It's been triaged correctly but nothing happens.You're not in an official leading position but you decide to act although you're not able to fix it yourself.
- Ask for **volunteers**. You can't give orders to specific people, of course. No success.
- Wait a little bit then ask **again**. Still no success. No one can / dares?
- Finally ask to **someone specifically**. He says OK.
- In some situations this would stop here, problem solved. But not in our example :)
- Several weeks or months later, notice there **still no progress** nor any new status on the bug report. Ask. The first volunteer finally tells he won't be able to do it in time (or can't be joined).
- Back to square one.
- It seems **no one wants to take reponsibility alone** for this difficult task, so you try to form a commando of willing contributers and work with them off-list so that they feel more involved.
- It works, you finally find a **solution together**.
- **This is a form of contribution that is not easily described in « How to contribute ? » documents, but it's really valuable and doesn't require extraordinary skills. Just communication and commitment.**

Case study: The QA and security teams

Both teams used to rely entirely on paid employees at Mandriva

- Transition to full community involvement.
- A real concern when we started: not the sexiest of tasks.

QA (Quality Assurance) team:

- Tests update candidates and ISOs.
- Lucky to find a great team leader who shaped the team.
- Great communication.
- Weekly meetings on IRC. Helps feeling like a crew.
- Efficient mentoring program: "go-to" team for new contributors.
- Can be a step towards packaging or other contribution forms.
- Tailor-made tools for the QA workflow.

Community QA team now as effective as the previous Mandriva QA labs (who worked on a subset of packages only).

Case study: The QA and security teams

Security team

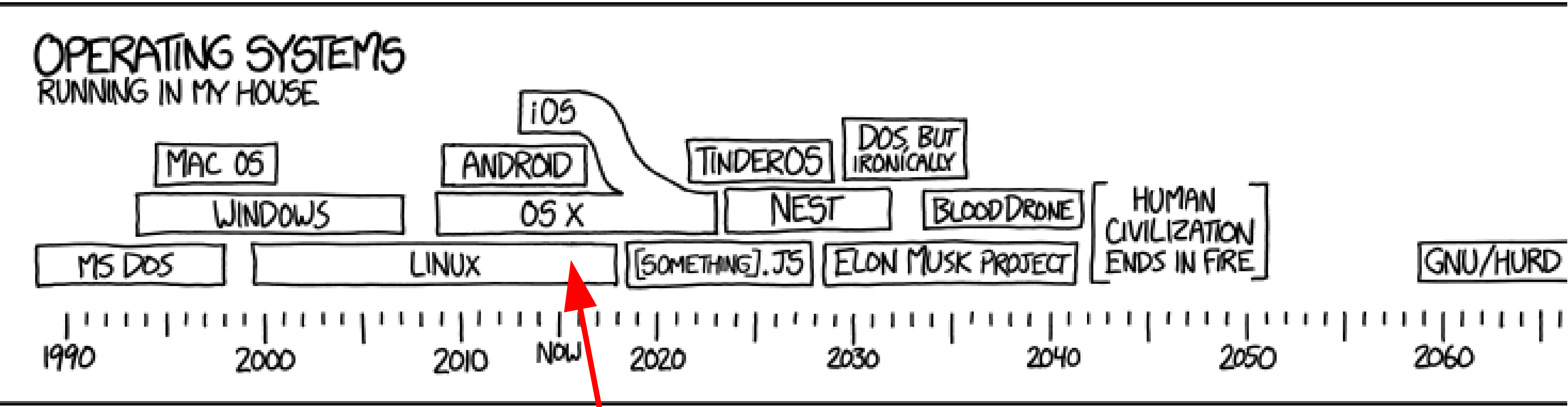
- One extremely dedicated contributor handling most of the security watch
- Difficulty to get packagers to fix security issues in a timely manner (even more if the package has no official maintainer)
- Same one contributor ends up doing the majority of the patching work

- Surprisingly, often as fast as distros who have more resources to handle security issues.
- High confidence in our updates thanks to QA testing.

→ Worked so far, but not sustainable
Additional packager focusing on security, there is hope :)

The way forward...

So what's now ???



xkcd.com

Infrastructure and tools : what about the original goals ?

What we wanted

- Start simple and clean, following the main project priorities (big delay after announcement)
- Infrastructure to enable the project, empower contributors
- Make it simple but strong... **6 years ago**

What we did

- Build infrastructure managing both resources and reliability (now more than it was in Mandriva)
- A small but motivated and experienced sysadmins team (Mandriva + technical skills)

Infrastructure and tools : empower the community

Reliable but far from being perfect

- Update an oldish infrastructure (finalize svn=>git migration)
- Adapt to current tool to fit with potential new/existing contributers
- A sysadmin is rare – lack of delegation (trust needed / missing granular rights)
- Patches management tool: mailing list / irc / Bugzilla

Let's increase our todo list

- Enforce existing workflow using 2017 tools : gitlab (easy access to code and packages), code review (make contribution easier)
- Create real project for Mageia upstream tools using a set of coherent tools to give more visibility
- Rethink our worklow properly : stop a svn-like use of git

We need your feedback

Feedback

Packagers organization : which model in other distributions ?

How to get people interested in bug fixing ?

How to get more contributions in non technical areas ? (design, communication)