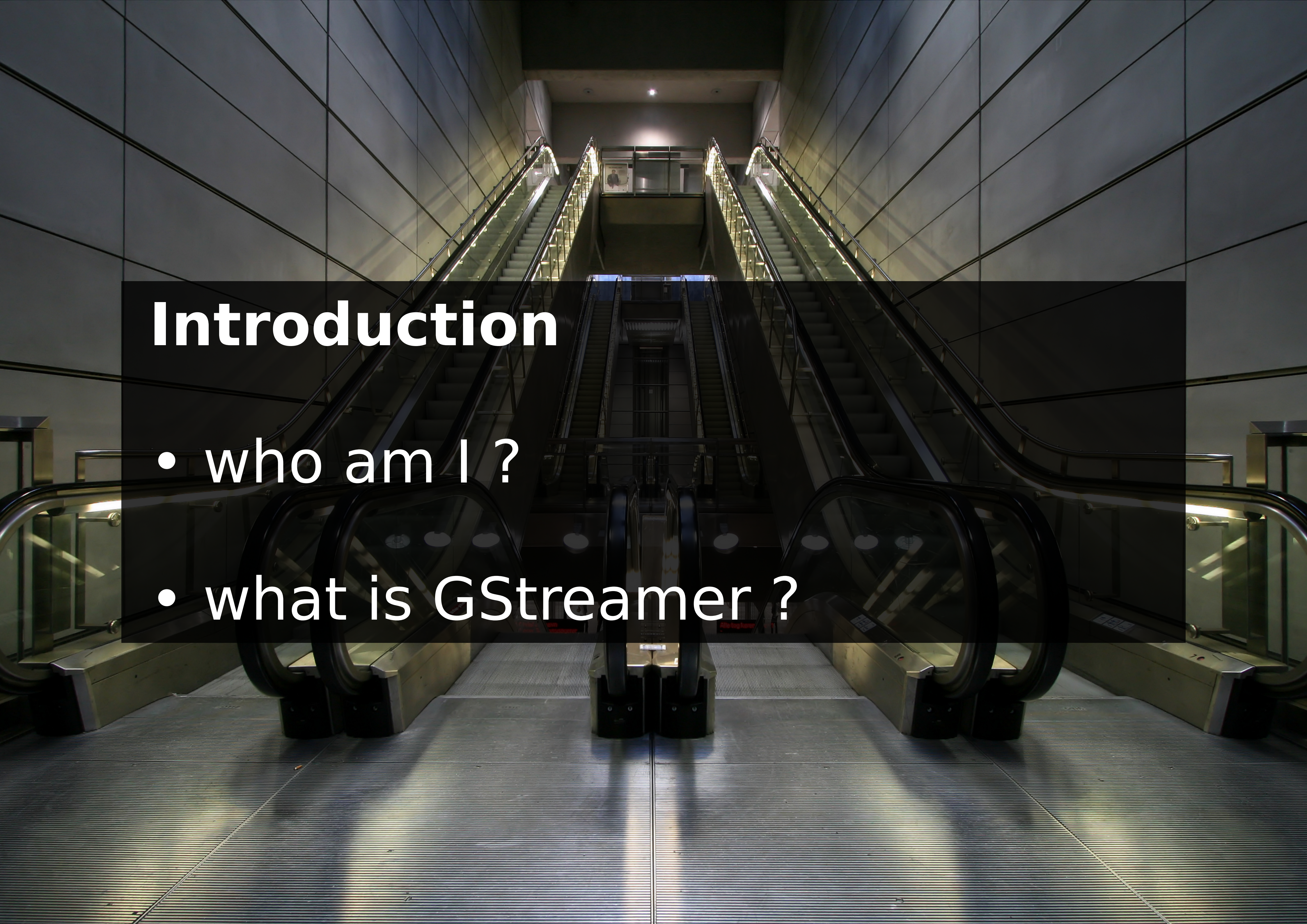
Taking GStreamer to the Next Level

GStreamer Conference, Edinburgh
22 October 2013

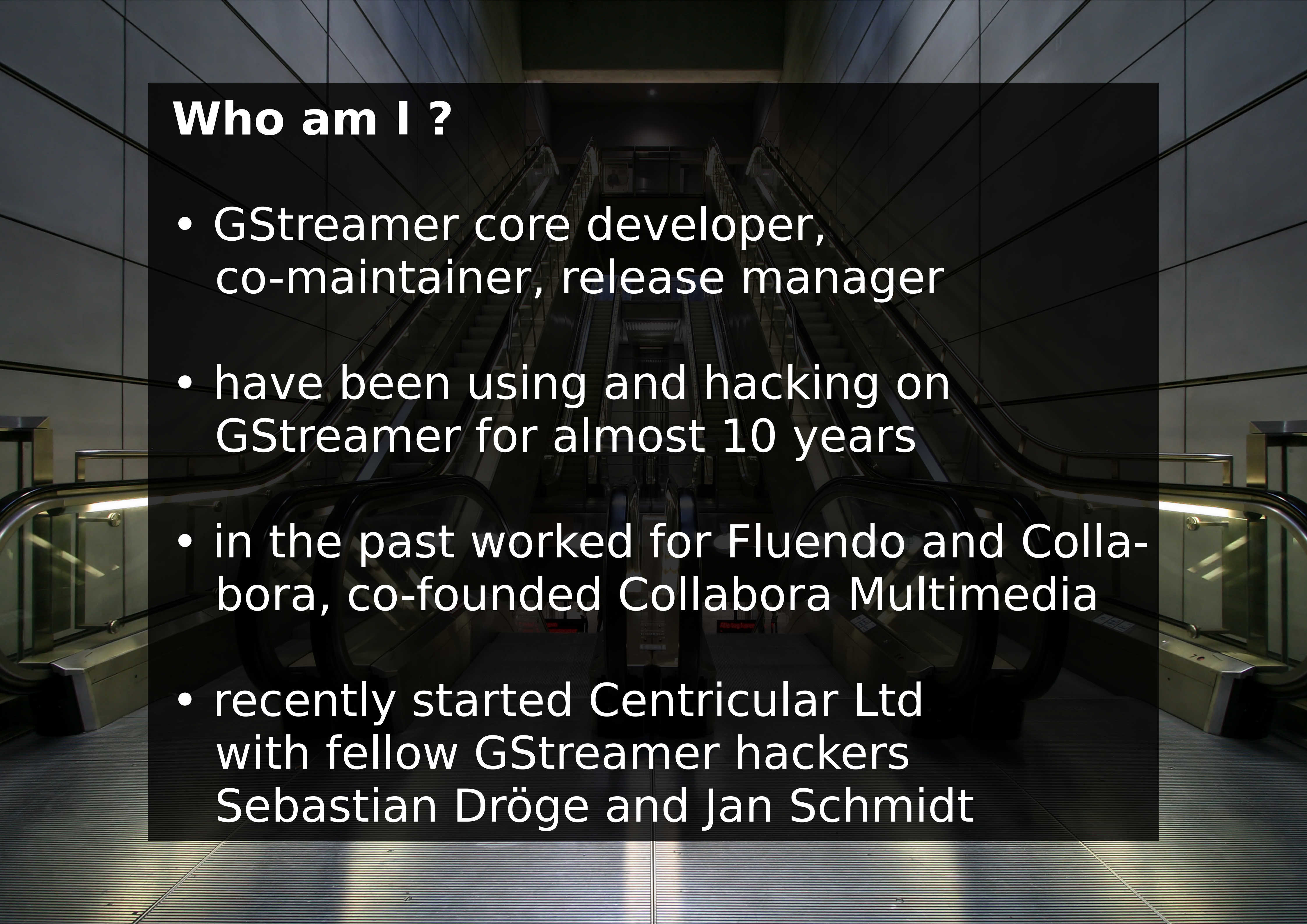
Tim-Philipp Müller <tim@centricular.com>

Centricular, www.centricular.com



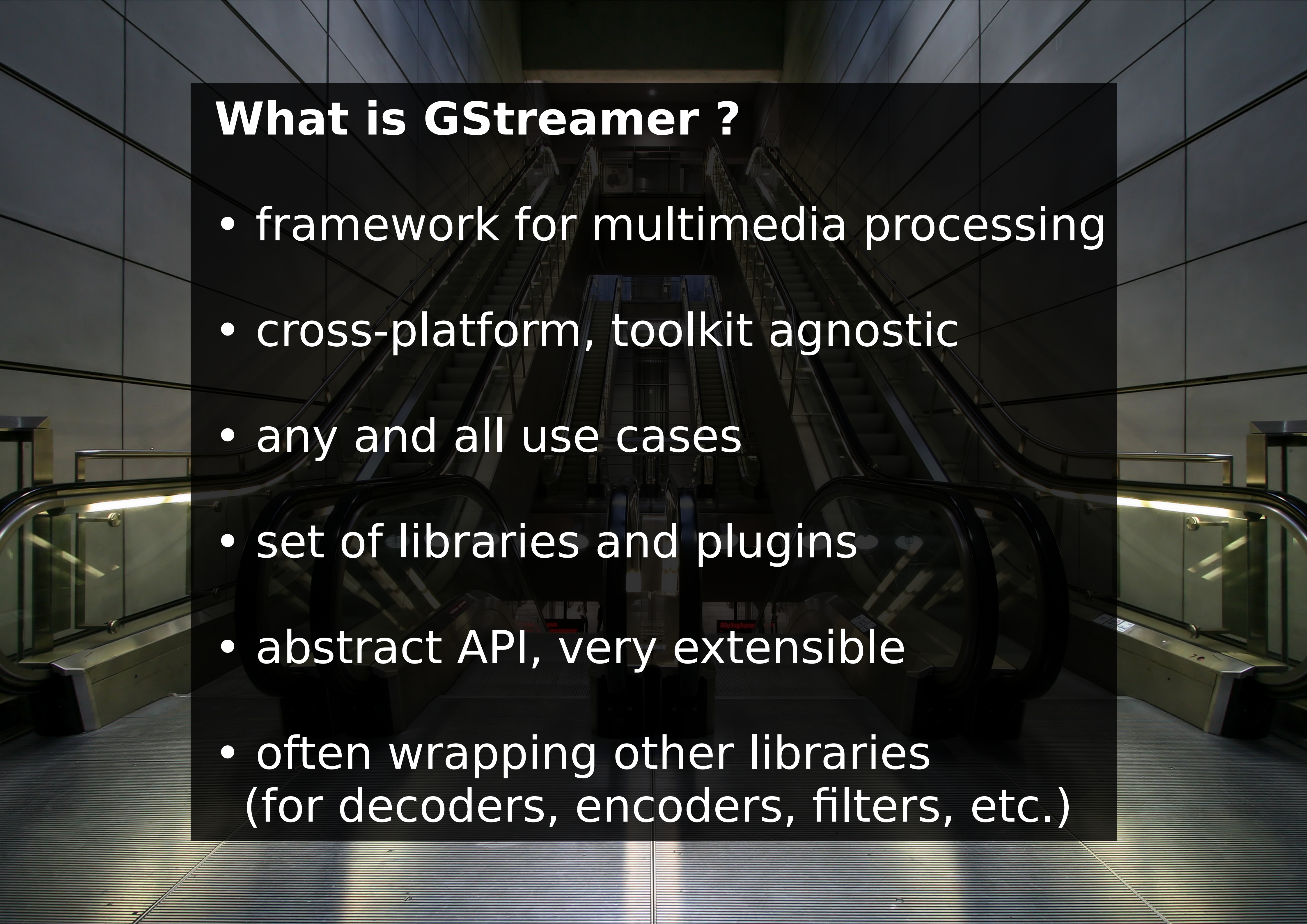
Introduction

- who am I ?
- what is GStreamer ?



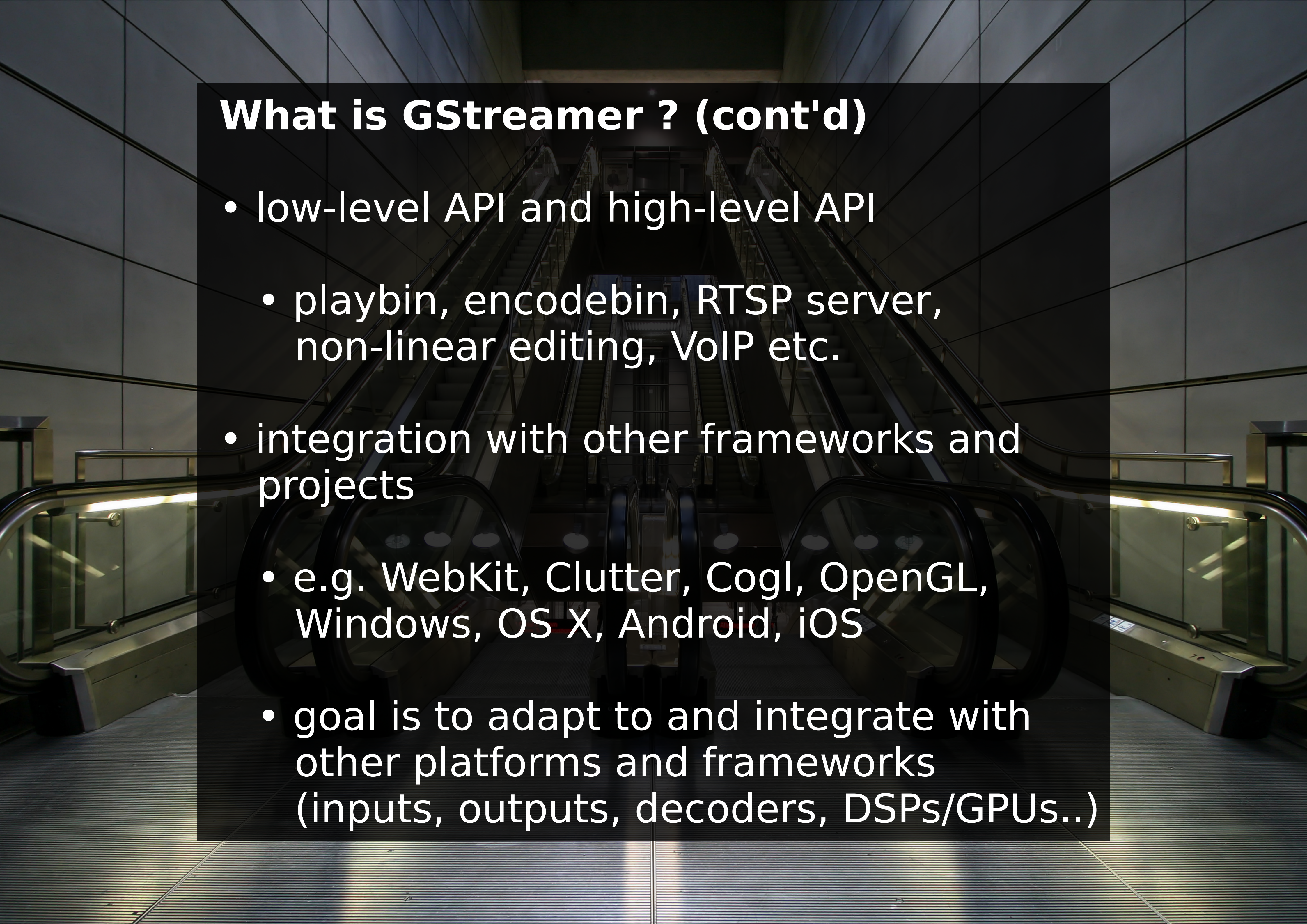
Who am I ?

- GStreamer core developer, co-maintainer, release manager
- have been using and hacking on GStreamer for almost 10 years
- in the past worked for Fluendo and Colla-bora, co-founded Collabora Multimedia
- recently started Centricular Ltd with fellow GStreamer hackers Sebastian Dröge and Jan Schmidt



What is GStreamer ?

- framework for multimedia processing
- cross-platform, toolkit agnostic
- any and all use cases
- set of libraries and plugins
- abstract API, very extensible
- often wrapping other libraries
  (for decoders, encoders, filters, etc.)

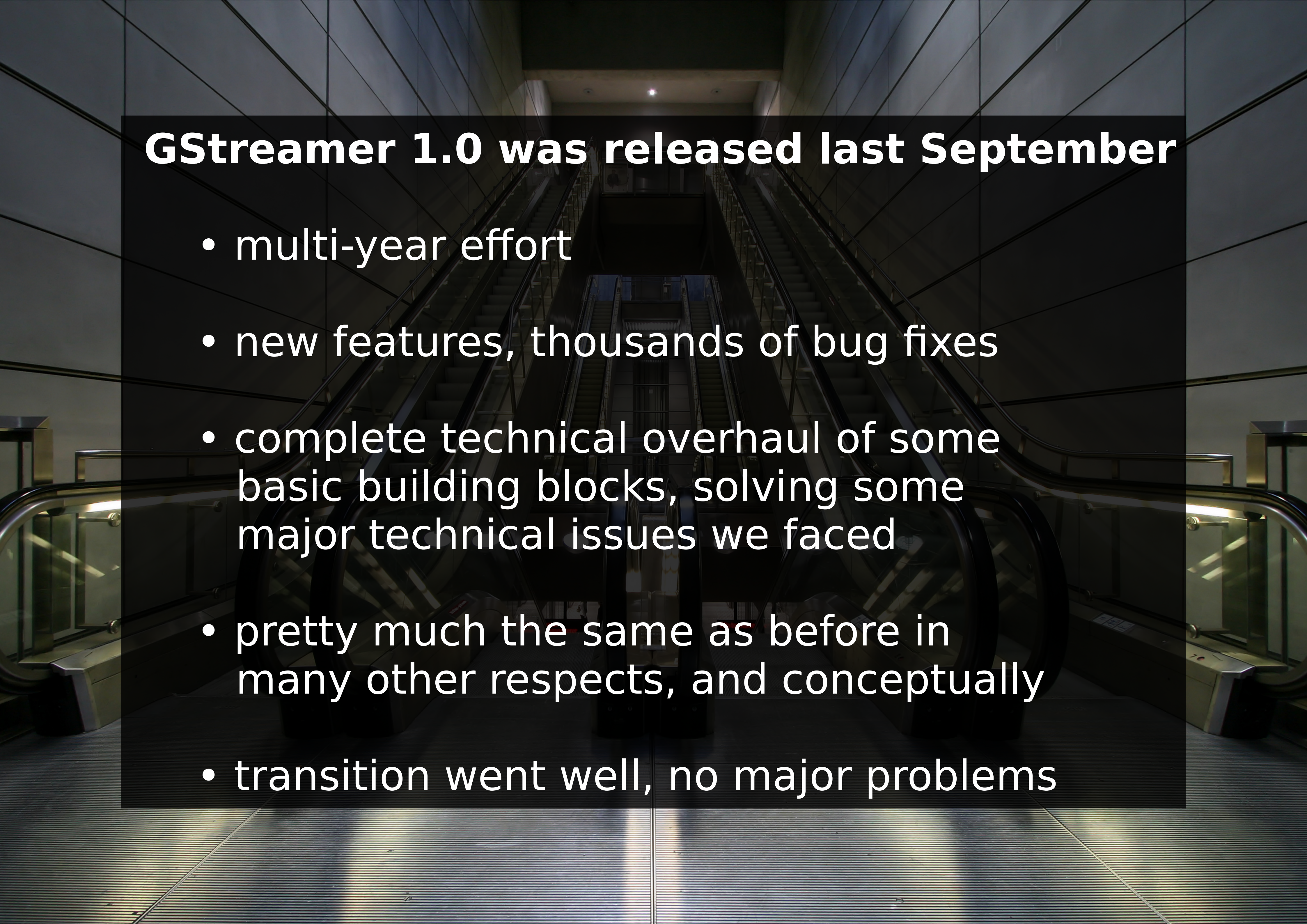


What is GStreamer ? (cont'd)

- low-level API and high-level API
  - playbin, encodebin, RTSP server, non-linear editing, VoIP etc.
- integration with other frameworks and projects
  - e.g. WebKit, Clutter, Cogl, OpenGL, Windows, OS X, Android, iOS
  - goal is to adapt to and integrate with other platforms and frameworks (inputs, outputs, decoders, DSPs/GPUs..)

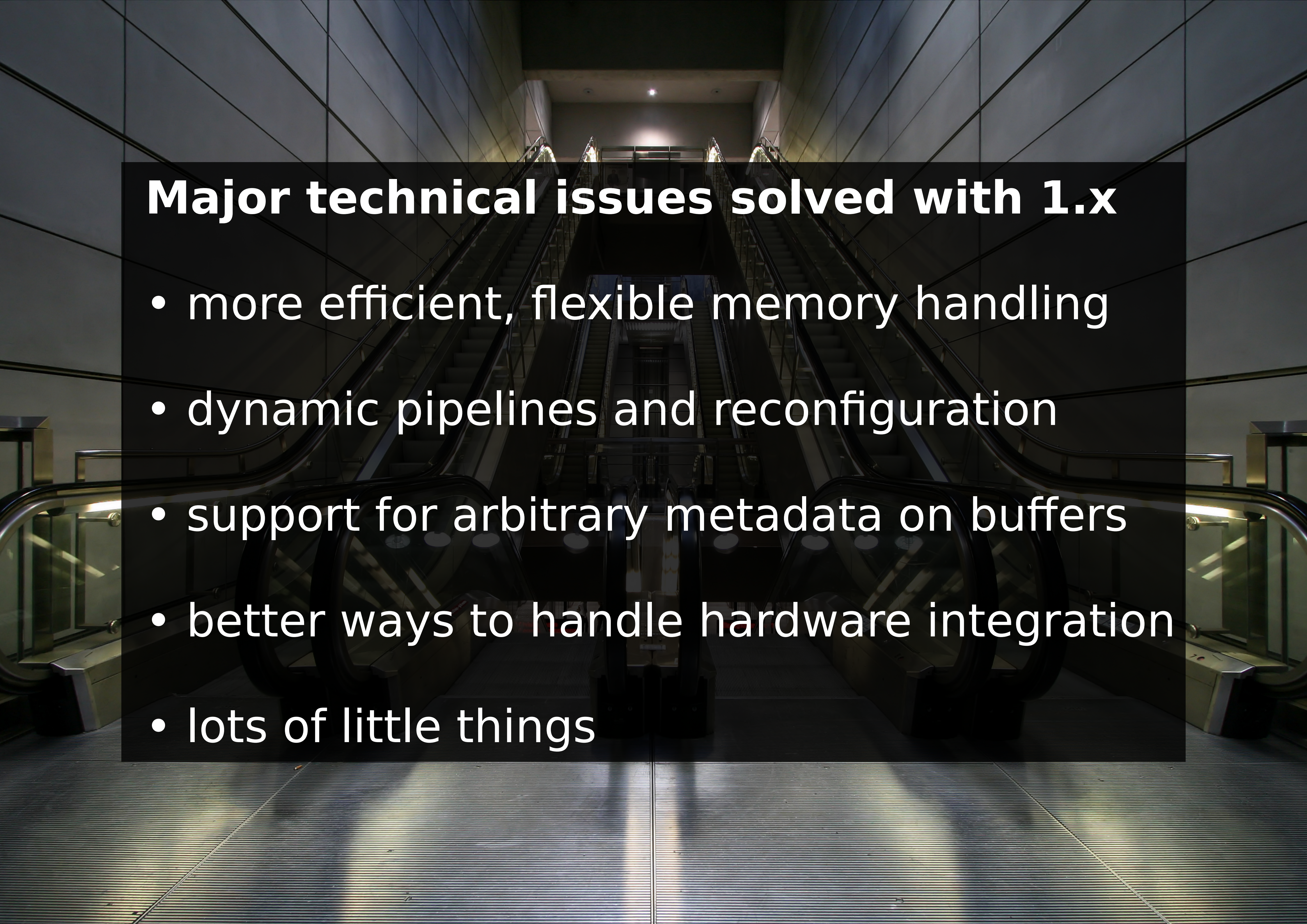
But why am I telling *you* what GStreamer is ?

We're at a unique point in the project's history



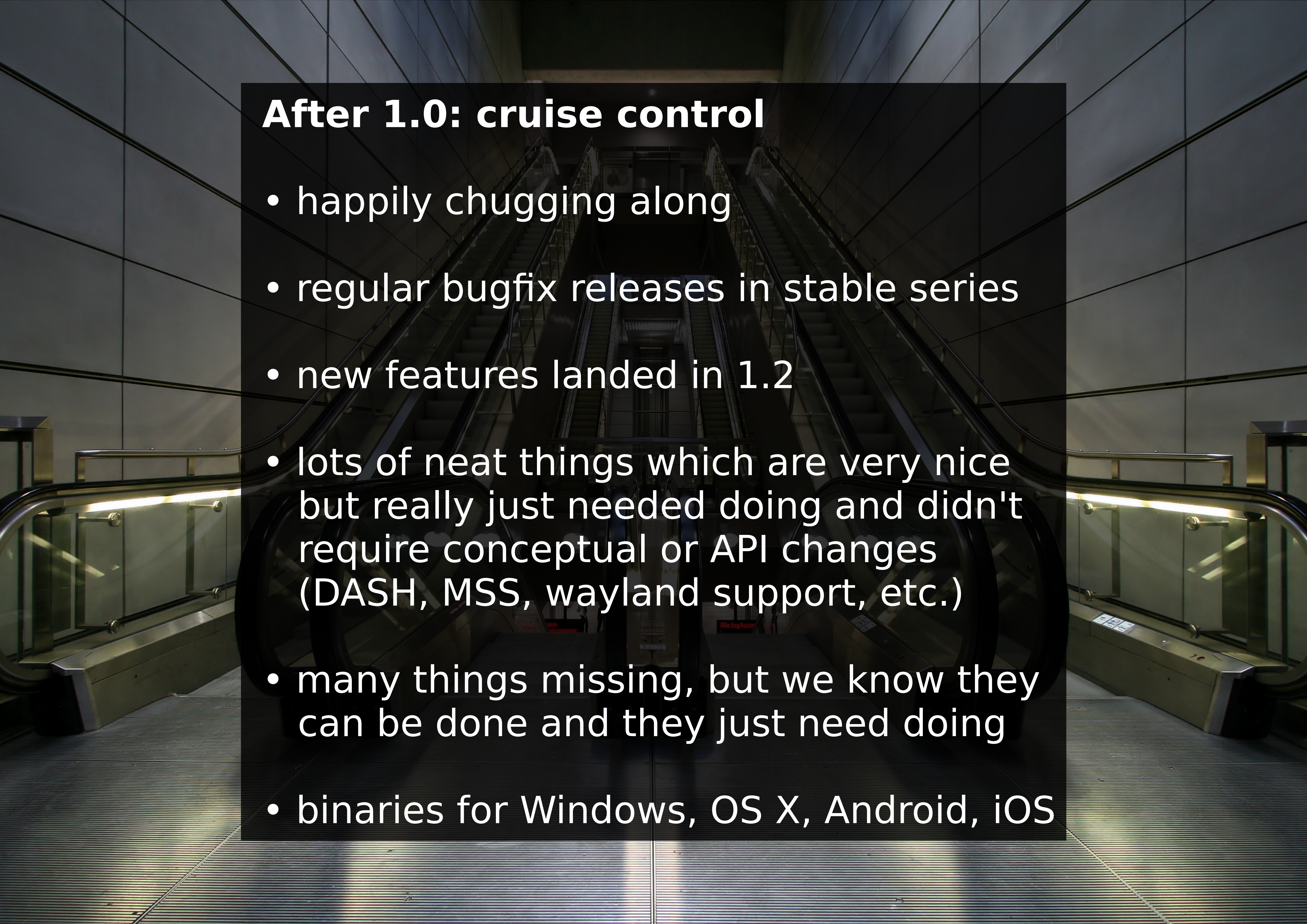
GStreamer 1.0 was released last September

- multi-year effort
- new features, thousands of bug fixes
- complete technical overhaul of some basic building blocks, solving some major technical issues we faced
- pretty much the same as before in many other respects, and conceptually
- transition went well, no major problems



Major technical issues solved with 1.x

- more efficient, flexible memory handling
- dynamic pipelines and reconfiguration
- support for arbitrary metadata on buffers
- better ways to handle hardware integration
- lots of little things



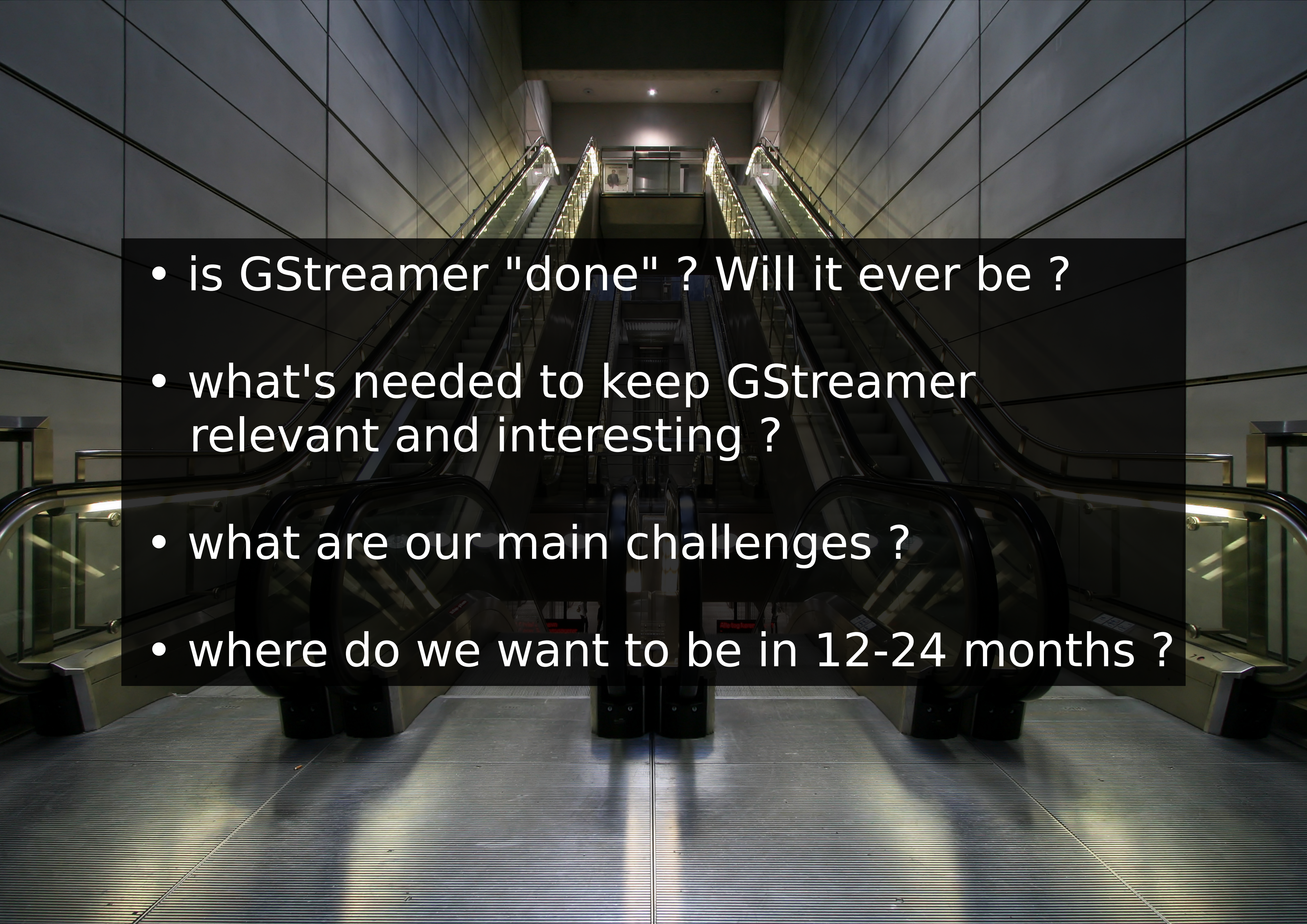
After 1.0: cruise control

- happily chugging along
- regular bugfix releases in stable series
- new features landed in 1.2
- lots of neat things which are very nice but really just needed doing and didn't require conceptual or API changes (DASH, MSS, wayland support, etc.)
- many things missing, but we know they can be done and they just need doing
- binaries for Windows, OS X, Android, iOS

Good time for some reflection

- what are our goals ?
- where are we at ?
- where do we want to go from here?

How do we take GStreamer to the Next Level ?

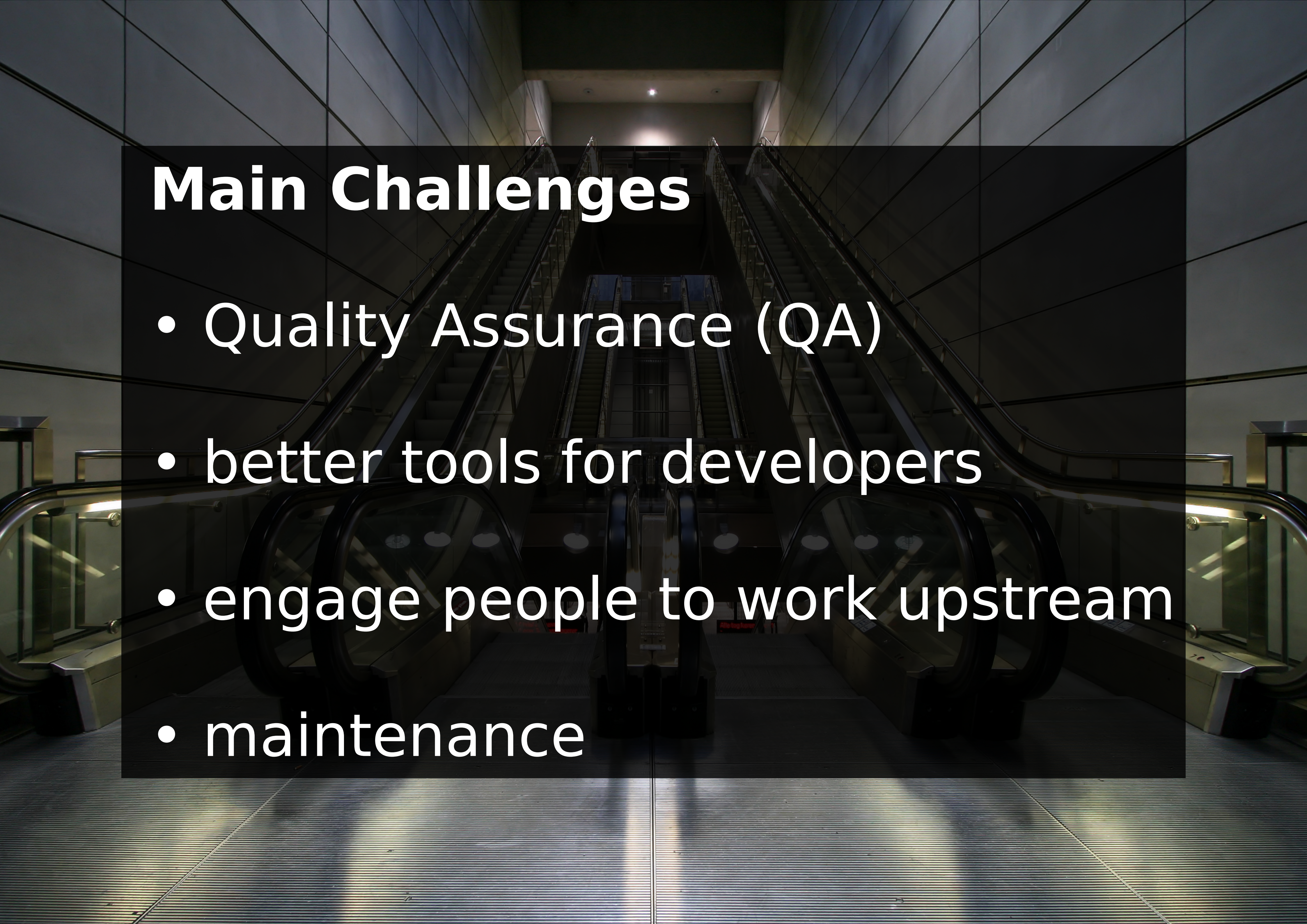


- is GStreamer "done" ? Will it ever be ?
- what's needed to keep GStreamer relevant and interesting ?
- what are our main challenges ?
- where do we want to be in 12-24 months ?

What major technical features are still missing ?

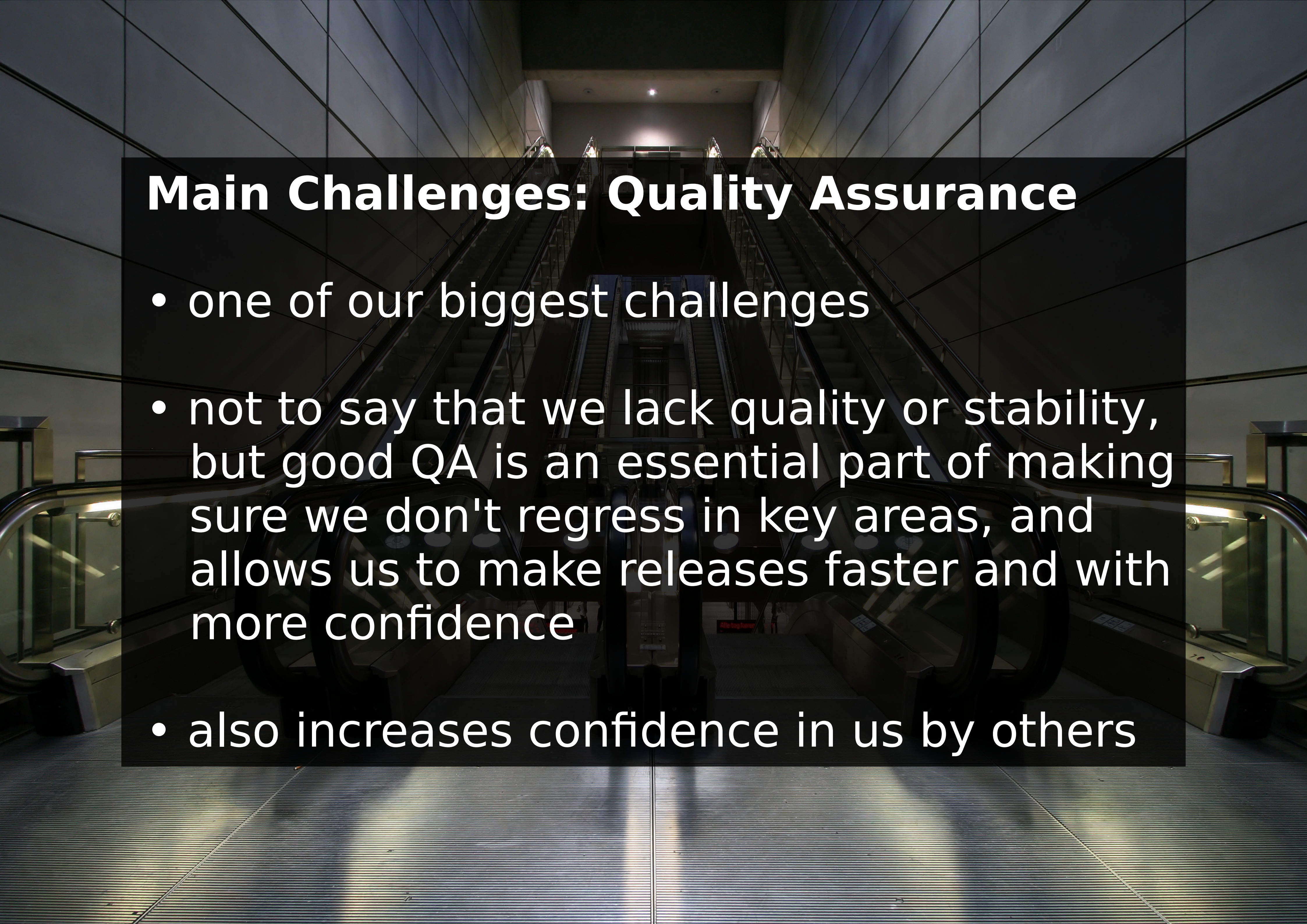
- sandboxing
- DRM
- 3-D video
- Blu-ray support
- end-to-end use of hardware acceleration across different layers/APIs
- nicer platform-specific higher-level APIs ?
- subtitle handling and rendering needs an overhaul

These will get done sooner or later.



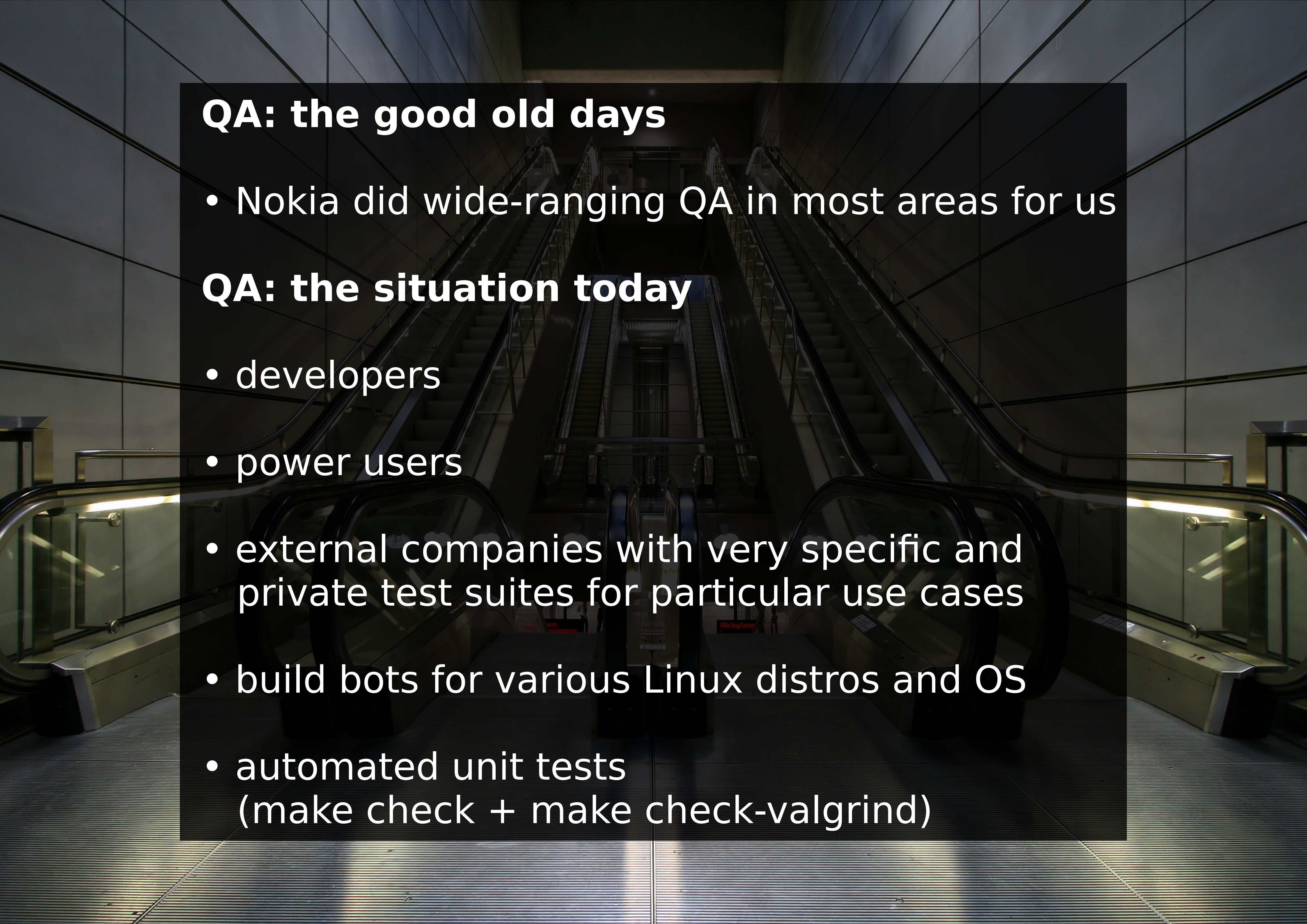
Main Challenges

- Quality Assurance (QA)
- better tools for developers
- engage people to work upstream
- maintenance



Main Challenges: Quality Assurance

- one of our biggest challenges
- not to say that we lack quality or stability, but good QA is an essential part of making sure we don't regress in key areas, and allows us to make releases faster and with more confidence
- also increases confidence in us by others

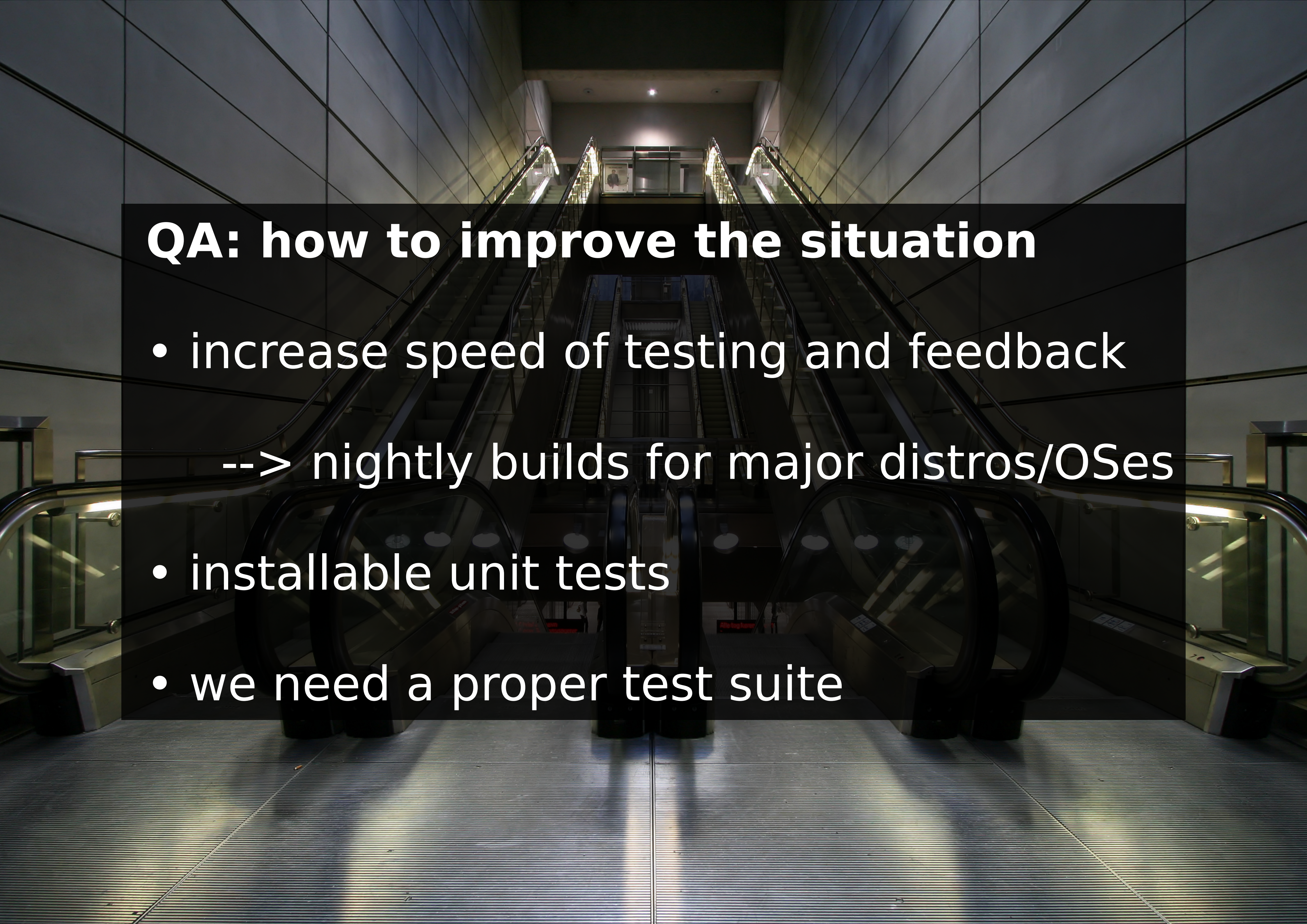


QA: the good old days

- Nokia did wide-ranging QA in most areas for us

QA: the situation today

- developers
- power users
- external companies with very specific and private test suites for particular use cases
- build bots for various Linux distros and OS
- automated unit tests (make check + make check-valgrind)

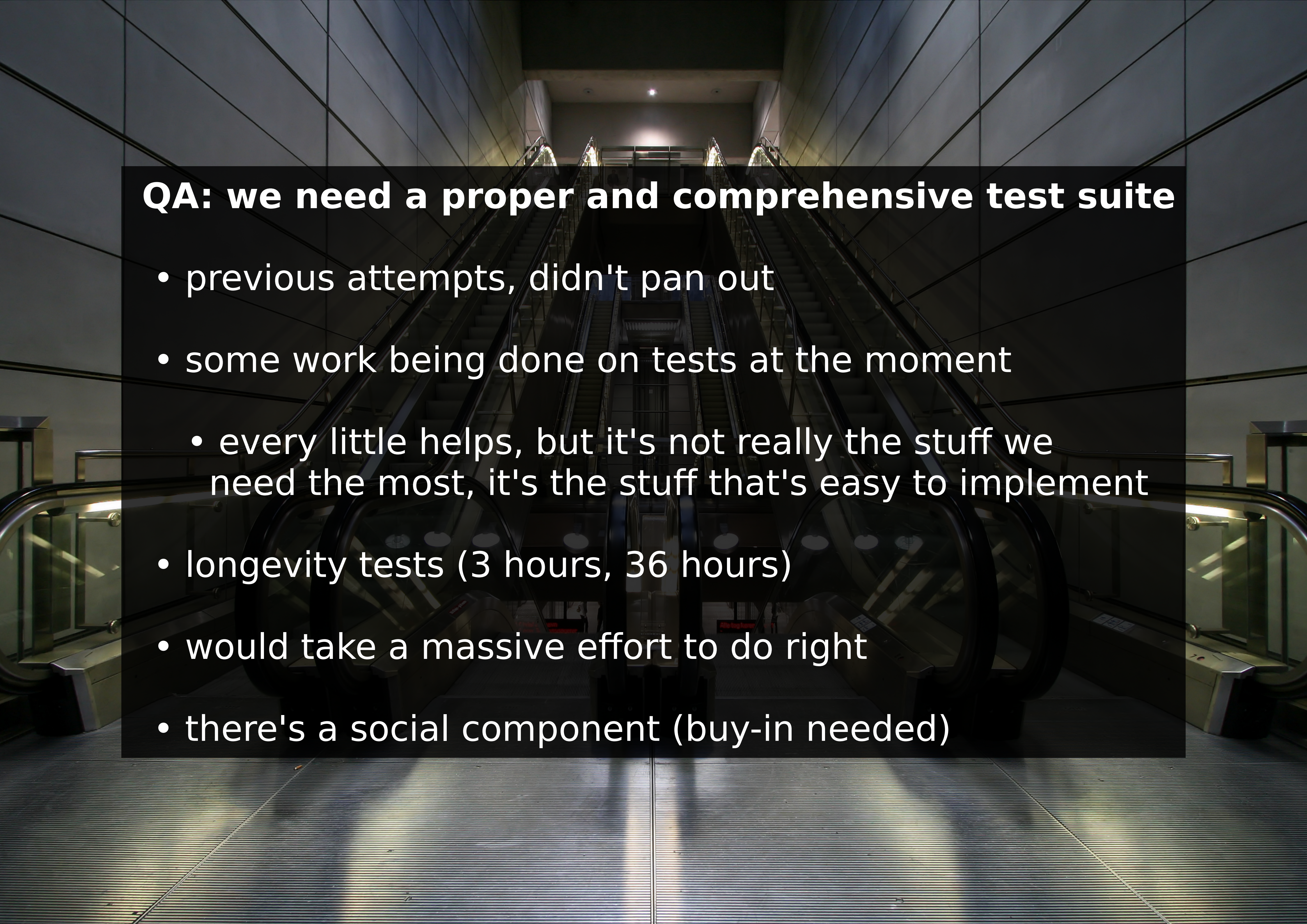


QA: how to improve the situation

- increase speed of testing and feedback

  --> nightly builds for major distros/OSes

- installable unit tests
- we need a proper test suite

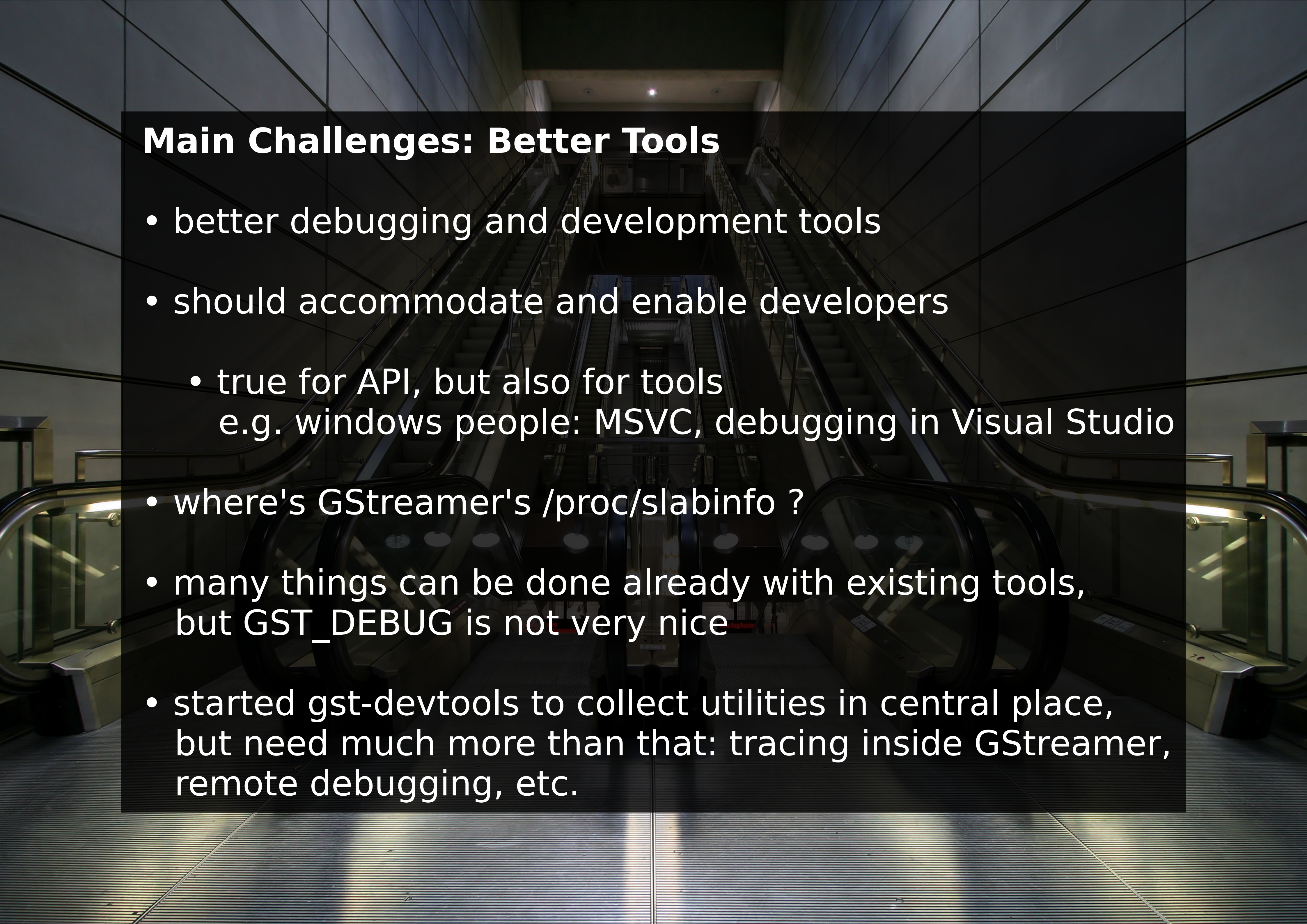


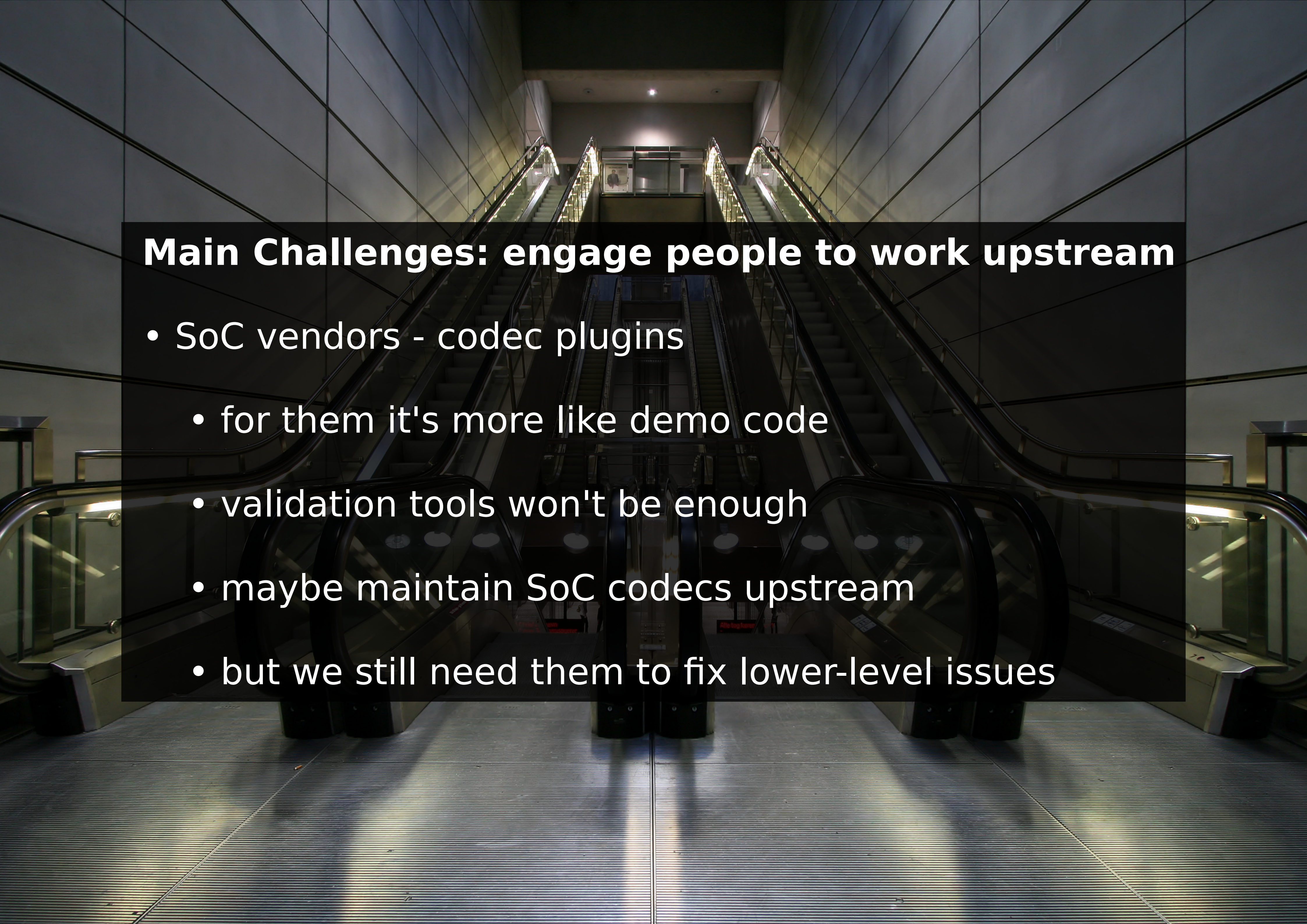
QA: we need a proper and comprehensive test suite

- previous attempts, didn't pan out
- some work being done on tests at the moment
  - every little helps, but it's not really the stuff we need the most, it's the stuff that's easy to implement
- longevity tests (3 hours, 36 hours)
- would take a massive effort to do right
- there's a social component (buy-in needed)

Main Challenges: Better Tools

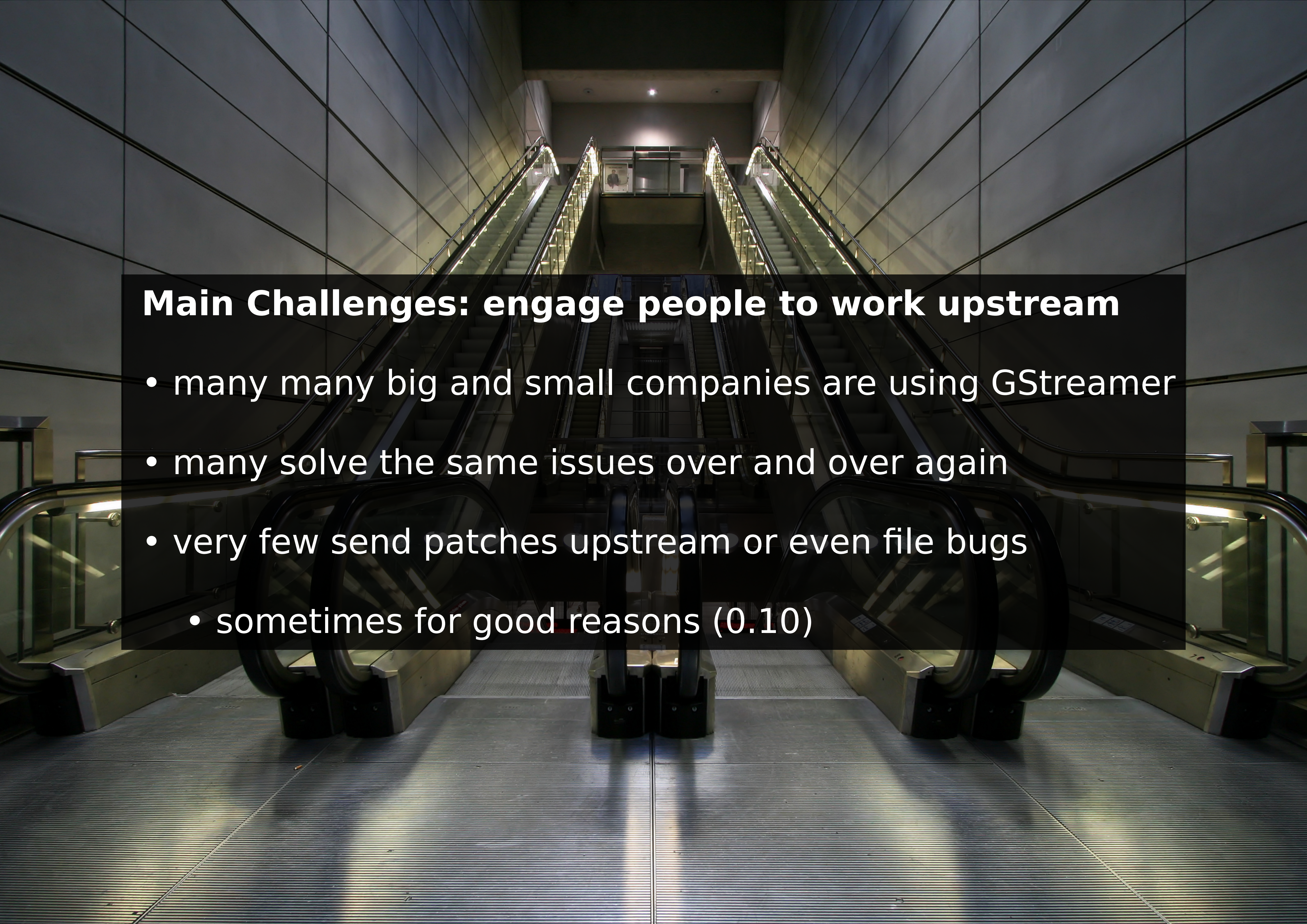
- better debugging and development tools
- should accommodate and enable developers
  - true for API, but also for tools
    e.g. windows people: MSVC, debugging in Visual Studio
- where's GStreamer's /proc/slabinfo ?
- many things can be done already with existing tools,
  but GST_DEBUG is not very nice
- started gst-devtools to collect utilities in central place,
  but need much more than that: tracing inside GStreamer,
  remote debugging, etc.





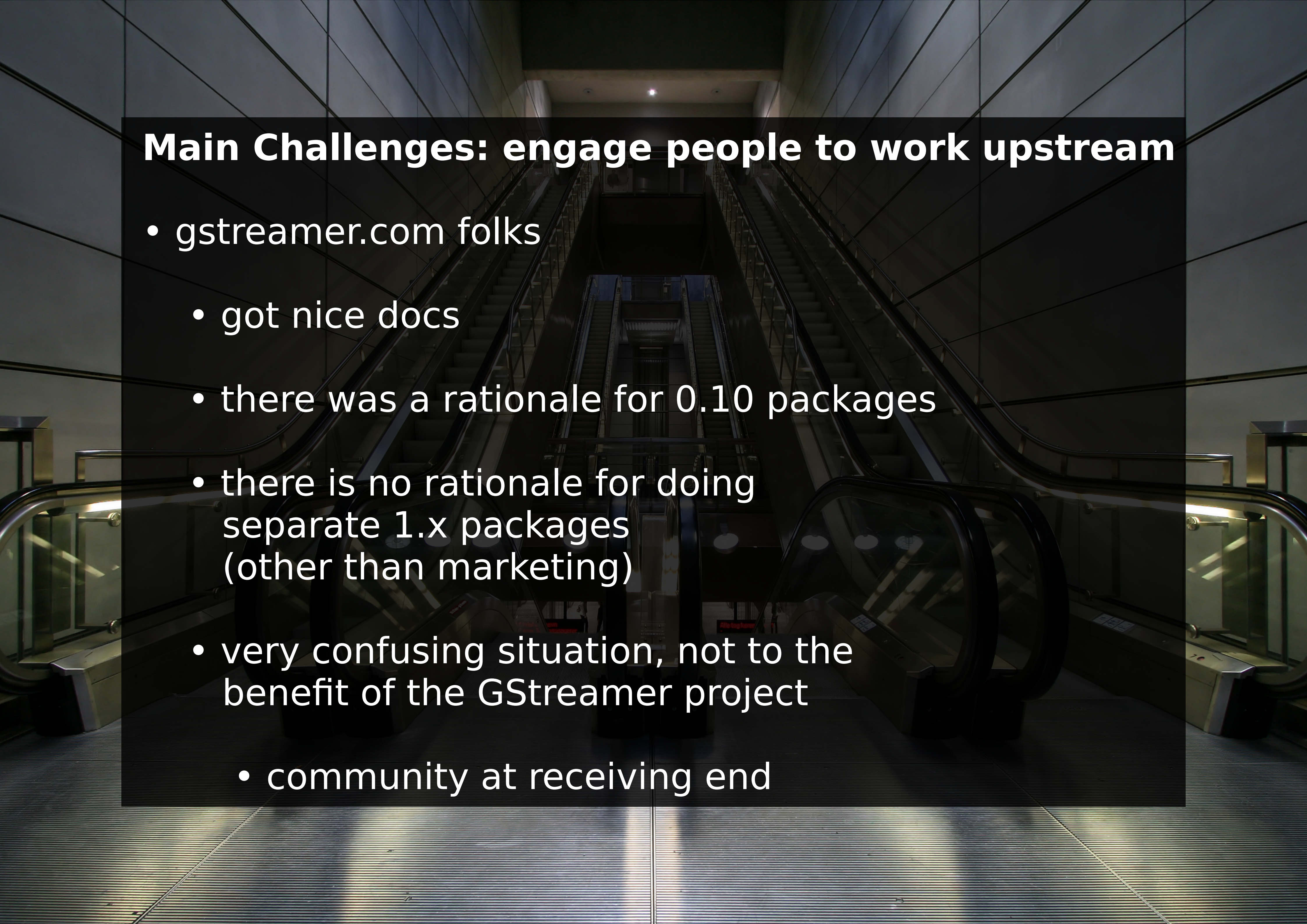
Main Challenges: engage people to work upstream

- SoC vendors - codec plugins
    - for them it's more like demo code
    - validation tools won't be enough
    - maybe maintain SoC codecs upstream
    - but we still need them to fix lower-level issues



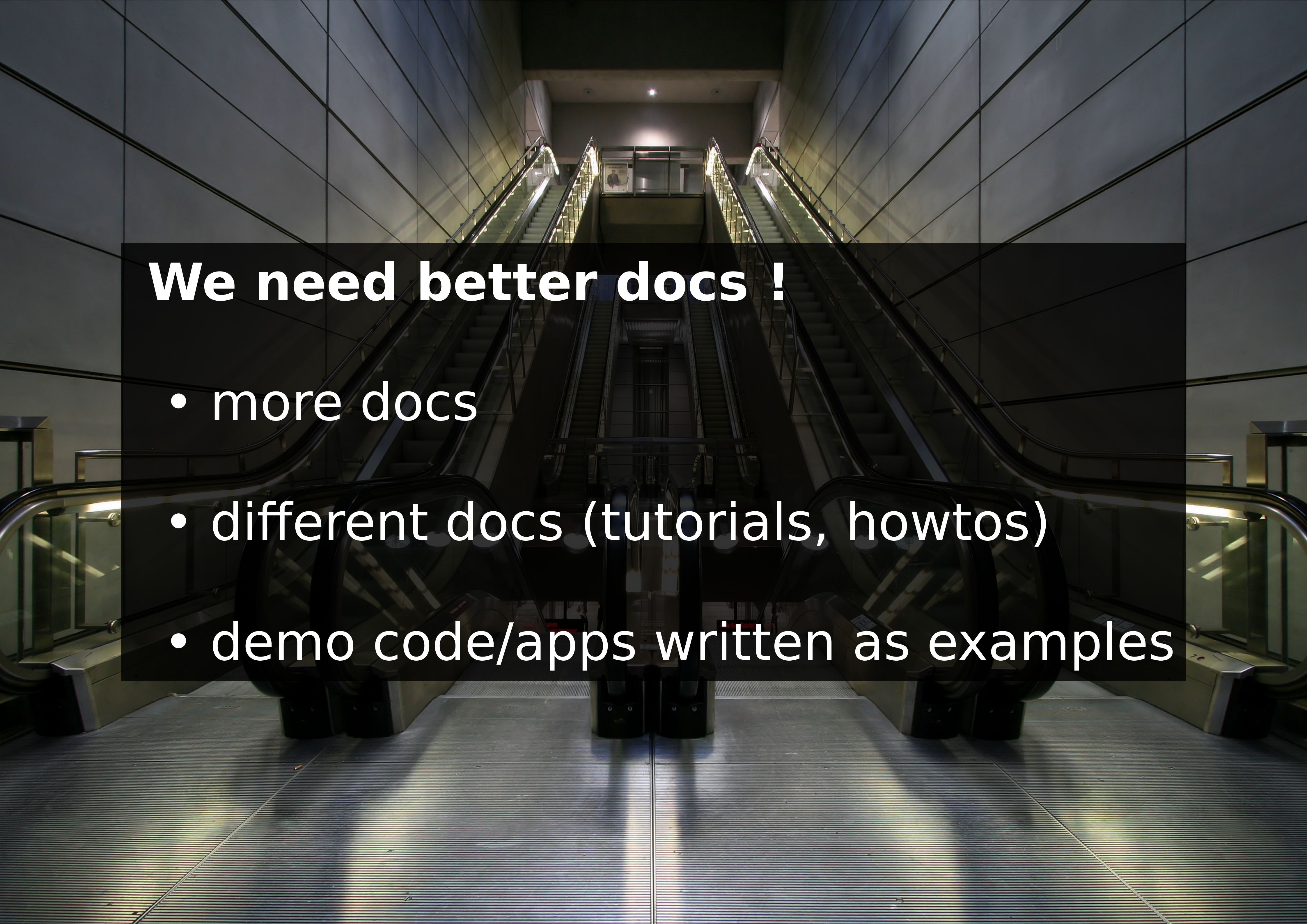
Main Challenges: engage people to work upstream

- many many big and small companies are using GStreamer
- many solve the same issues over and over again
- very few send patches upstream or even file bugs
  - sometimes for good reasons (0.10)



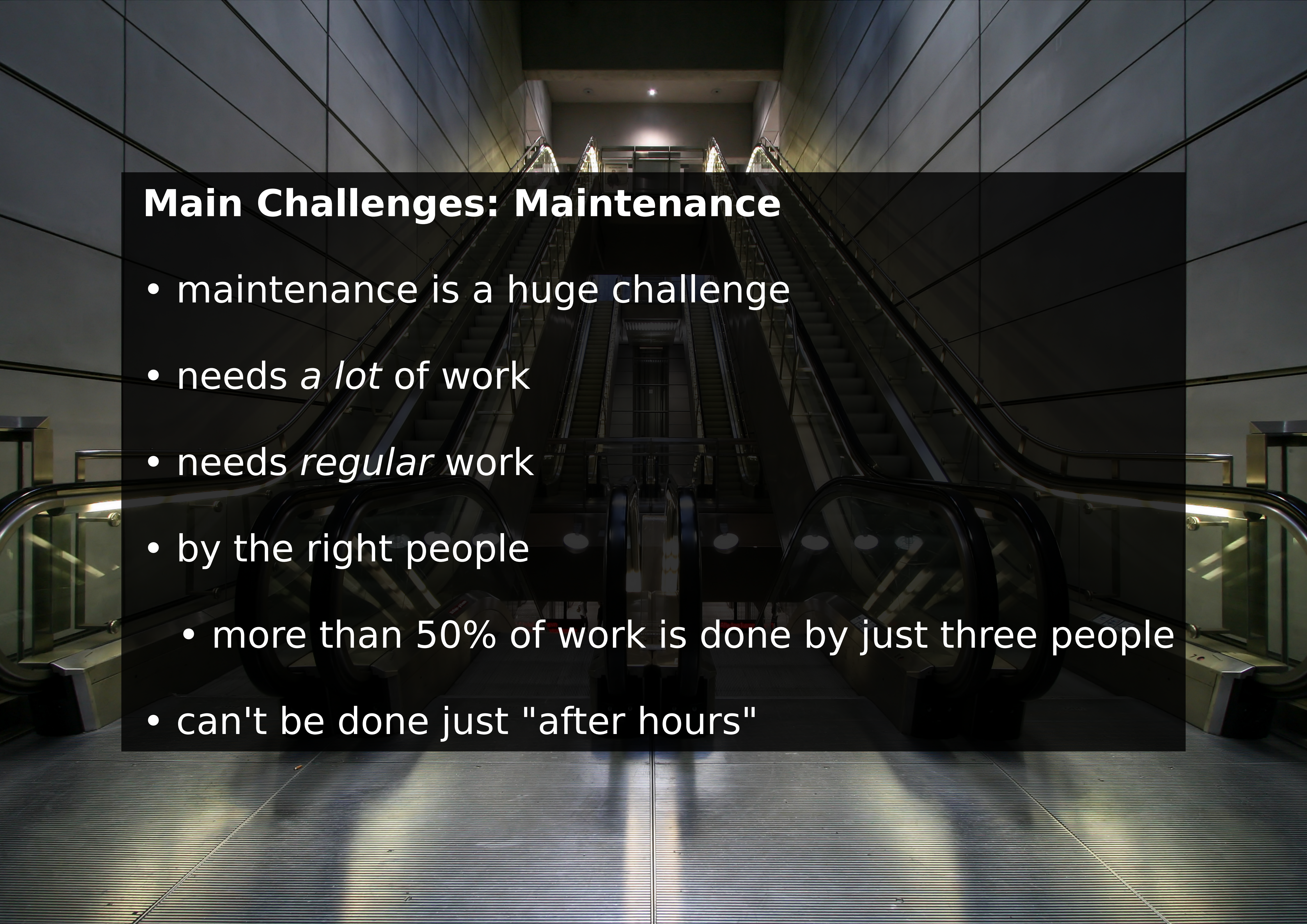
Main Challenges: engage people to work upstream

- gstreamer.com folks
  - got nice docs
  - there was a rationale for 0.10 packages
  - there is no rationale for doing separate 1.x packages (other than marketing)
  - very confusing situation, not to the benefit of the GStreamer project
    - community at receiving end



We need better docs !

- more docs
- different docs (tutorials, howtos)
- demo code/apps written as examples



Main Challenges: Maintenance

- maintenance is a huge challenge
- needs *a lot* of work
- needs *regular* work
- by the right people
  - more than 50% of work is done by just three people
- can't be done just "after hours"

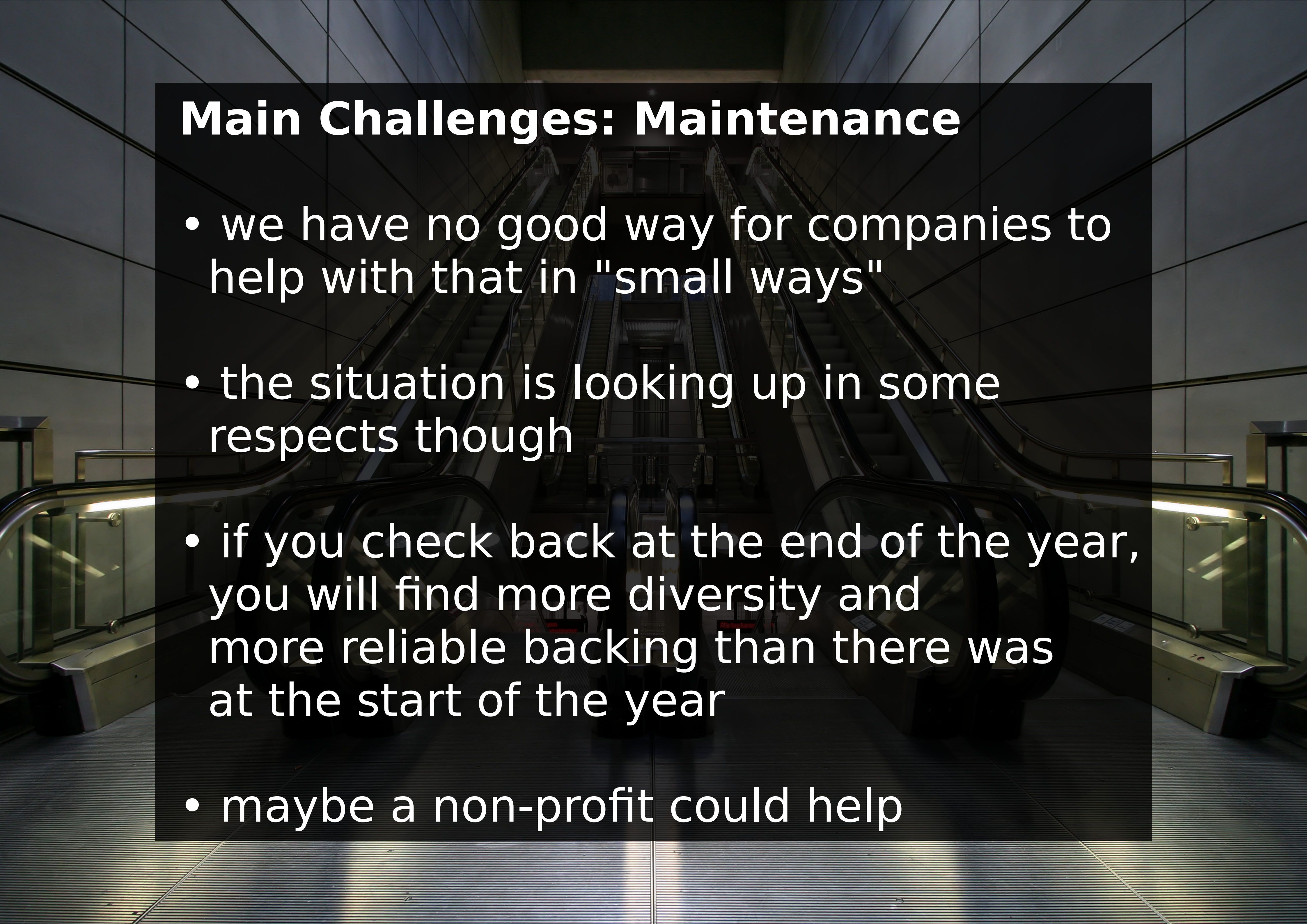
Main Challenges: Maintenance

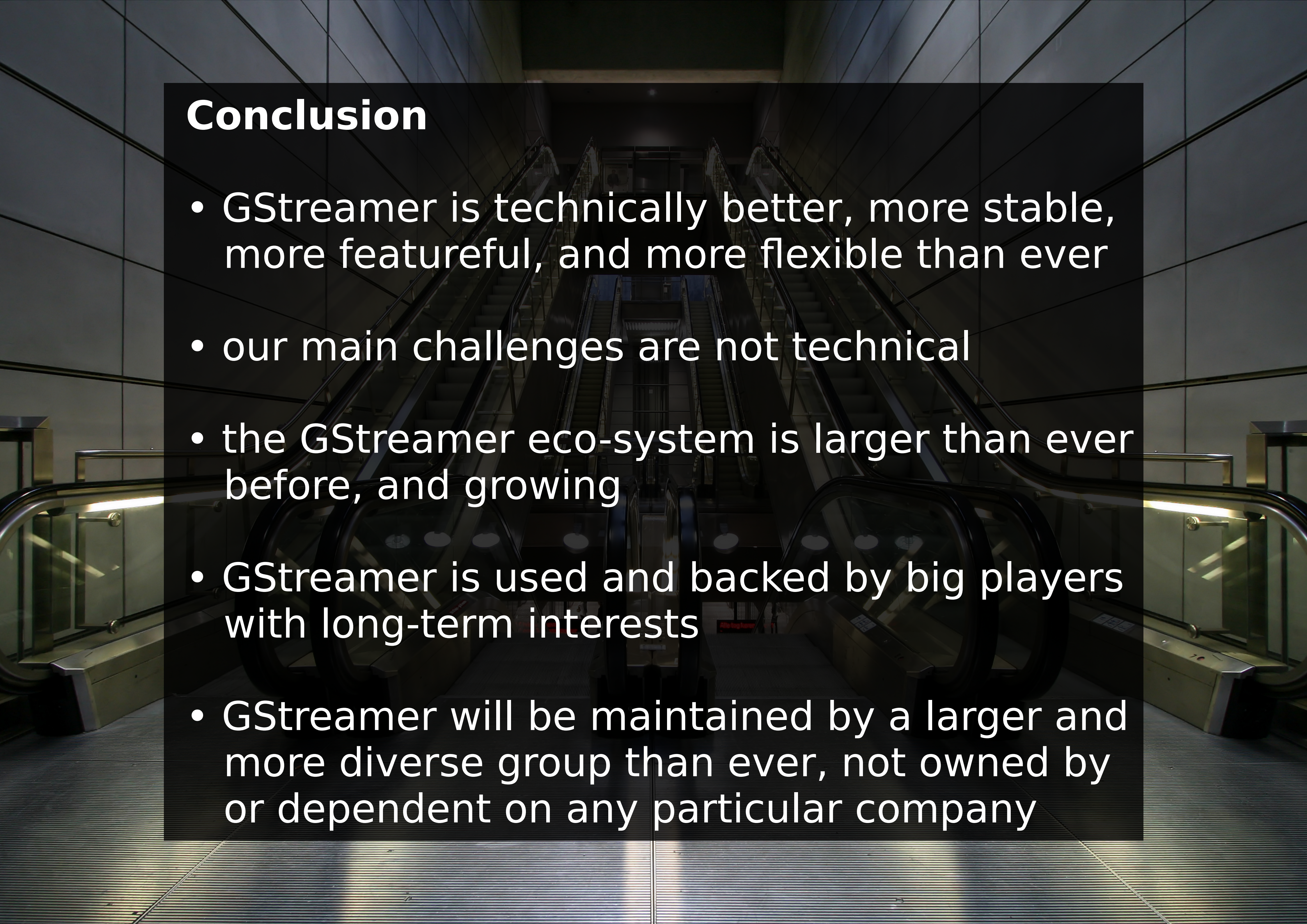
- we have no good way for companies to help with that in "small ways"

- the situation is looking up in some respects though

- if you check back at the end of the year, you will find more diversity and more reliable backing than there was at the start of the year

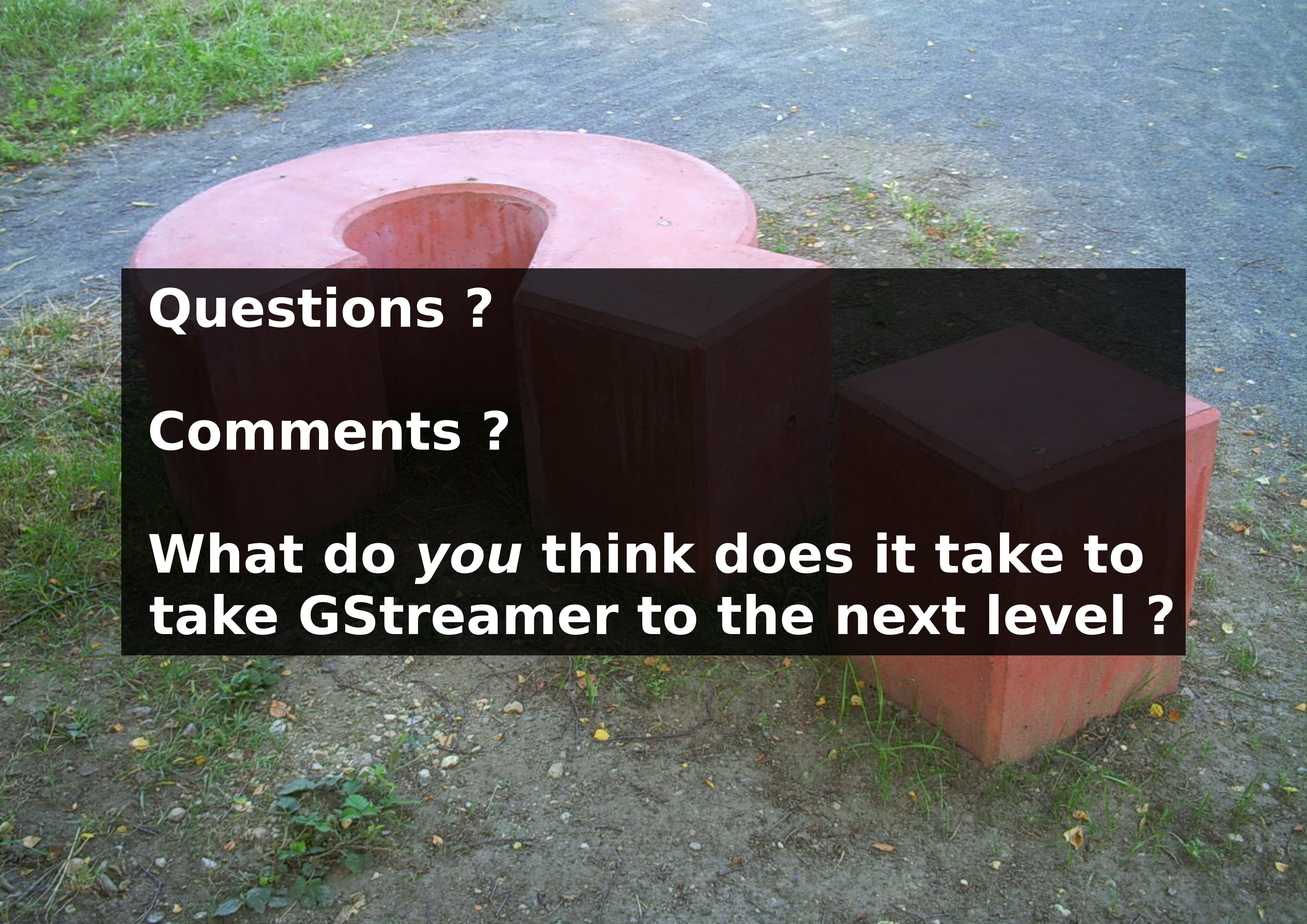
- maybe a non-profit could help





Conclusion

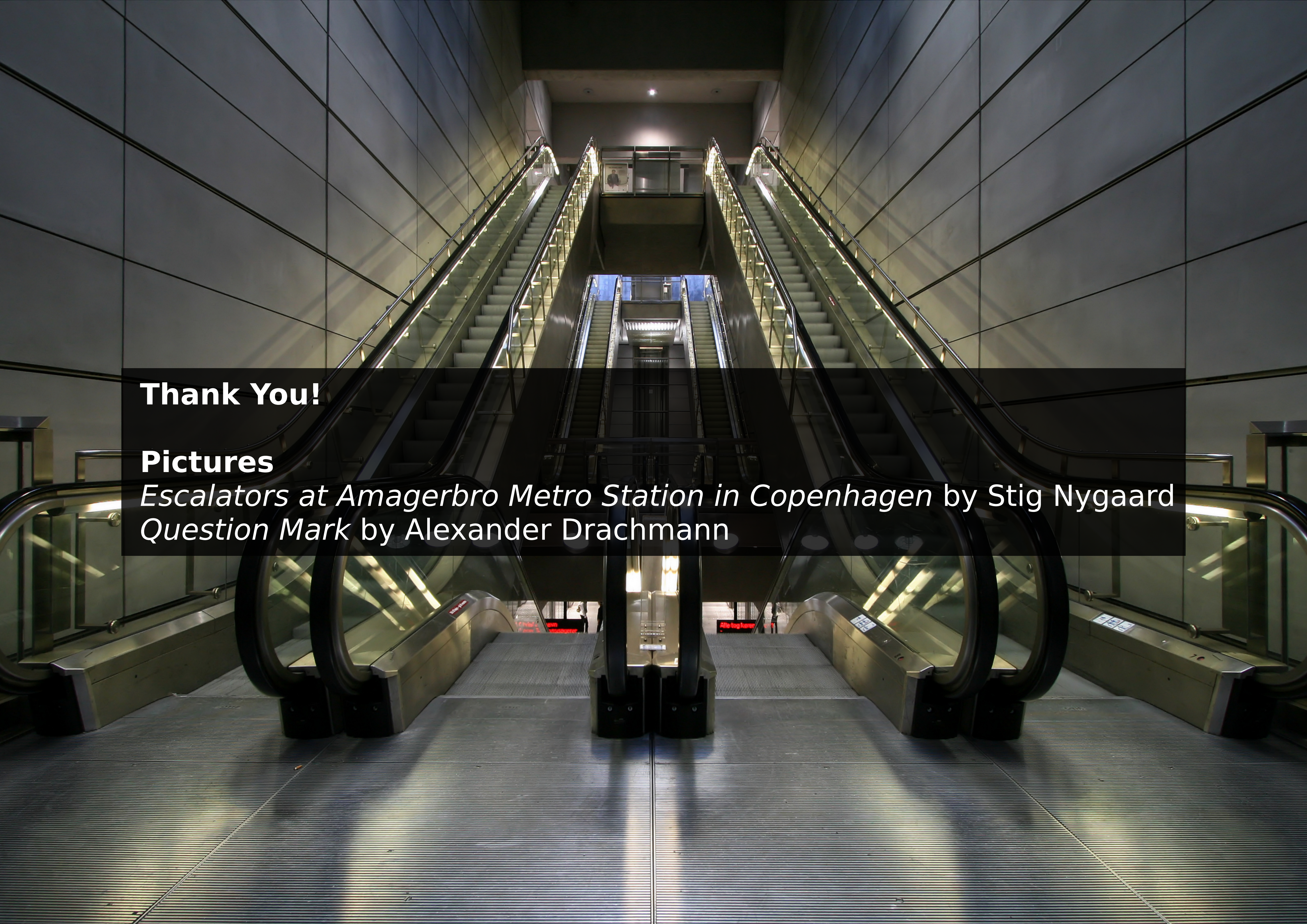
- GStreamer is technically better, more stable, more featureful, and more flexible than ever
- our main challenges are not technical
- the GStreamer eco-system is larger than ever before, and growing
- GStreamer is used and backed by big players with long-term interests
- GStreamer will be maintained by a larger and more diverse group than ever, not owned by or dependent on any particular company



Questions ?

Comments ?

What do *you* think does it take to take GStreamer to the next level ?



Thank You!

Pictures
Escalators at Amagerbro Metro Station in Copenhagen by Stig Nygaard
Question Mark by Alexander Drachmann