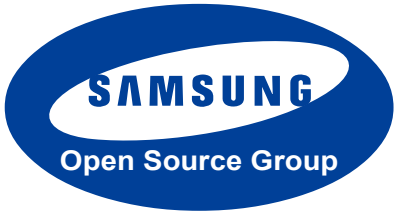
FFmpeg – A Retrospective

Reynaldo H. Verdejo Pinochet <reynaldo@osg.samsung.com>
Open Source Group / Samsung Research America / SV

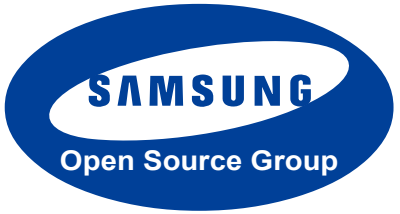
Introduction

About myself



- Senior Multimedia Software Engineer for the OSG (SRA/SV)
- Long time FOSS developer: MPlayer, FFmpeg, GStreamer, etc

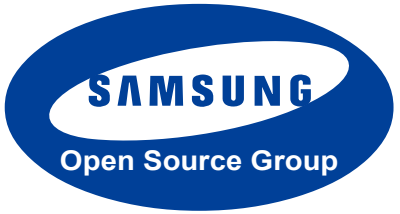
Agenda



- What is FFmpeg?
- A brief look at 15 years of project history
- Current Challenges
- Lessons Learned

What is FFmpeg?

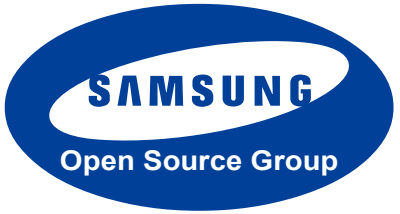
What is FFmpeg?



On the label

- Complete cross-platform solution to record, convert, and stream media

What is FFmpeg?

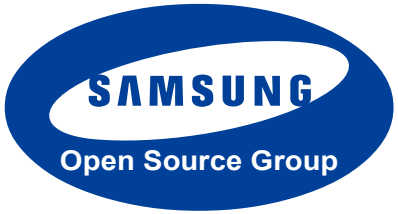


For real

- A collection of SW implementations of a huge number of codecs and de/muxers and a low level API for multimedia processing
- That and ***tools***

"FFmpeg is geared towards traditional use cases but since it is low level, it doesn't limit application scope."

What is FFmpeg?

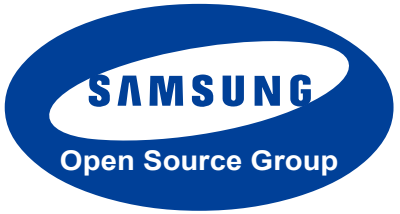


Licensing

Depending on what modules are enabled (--enable-gpl)

- LGPL 2.1+
- GPL 2+

What is FFmpeg?

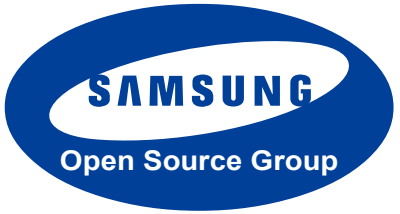


Community

Time for a risky statement ...

"It's a large, international free software development community that **shouldn't work** but — somehow — it **does**."

What is FFmpeg?

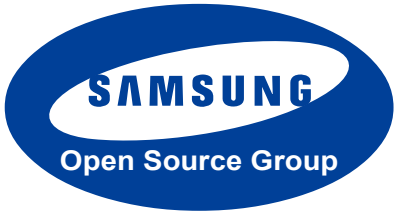


The previous statement might be hard to grasp but we will see later how this seemingly ***unique*** characteristic is:

- Not really that unique
- An interesting subject to discuss and learn from

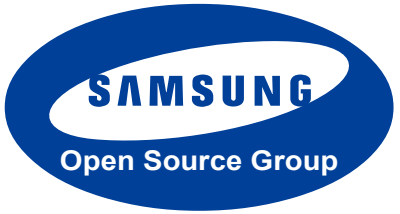
15 Years of Project History

The First Days



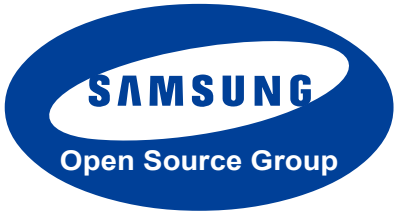
- FFmpeg was created 15 years ago by ***Gerard Lantau*** (a pseudonym for F.B.)
- First ***release*** made **December 20, 2000**
- Pretty much a single community with MPlayer for a long time. Both communities shared a particularly rude approach to PR

The First Days



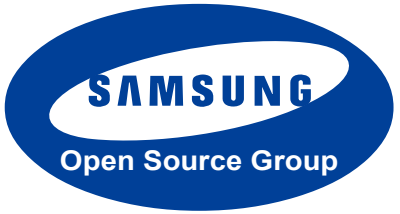
"'The MPlayer gang seems to relish nothing more than belittling their users and reminding them of just how little they know about Linux and computing in general" (Joe Barr for LinuxWorld, 2001) ***"Software by Marquis de Sade"***"

A Few Years In



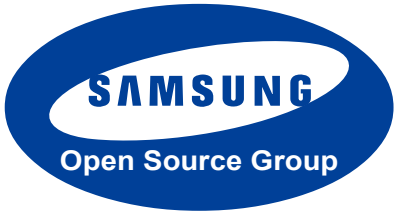
- 2003 H264 Decoder and Demuxer (Final draft of version 1 of the standard made circa May 2003)
- The team was already working on rolling their own codecs
  - FFV1 (intra only, lossless), circa 2003
- 2004, ***Michael Niedermayer*** gets the leadership from ***Fabrice Bellard*** after ∽ **3 years as a contributor**
  - No project lead change since then (but some on leadership style)

A Few Years In



- 2004, developers coming from related projects start showing up to stay
- 2005 Vorbis Decoder (Independent implementation)
- Up to 2006 there was a fairly slowly paced development speed (~100 c/m average) with a community of < 30 developers

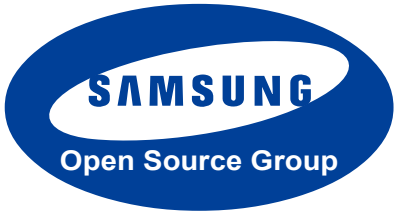
Growing pains



- As codebase matures, so do some egos
- Community continued making it to the front page
  - **"2010, FFmpeg's native VP8 decoder faster than google's libvpx"**
- By 2010 there was a general feeling of unrest regarding:
  - Merging of too-intrusive but otherwise important changes (MT)
  - No well defined set of rules on project managment nor community interaction
  - Project lead not communicating properly

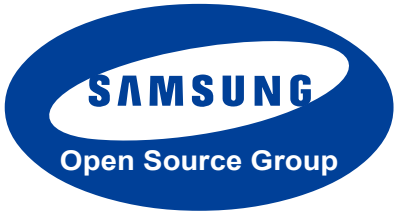
The Coup

The Coup



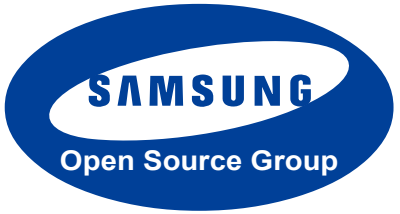
- A non trivial group of developers (18) attempts to perform a takeover. The idea was to kick out the current Maintainer and proceed from there with a new set of drafted rules
- Jan 18, 2011, **"[FFmpeg-devel] [ANNOUNCE] New FFmpeg maintainership"**

The Coup



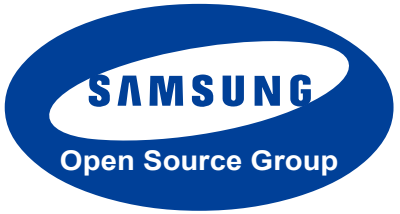
- An important detail was missing. They had infrastructure control but lacked domain control
- The situation led to the takeover attempt failing as such and becoming a fork, albeit an important one

The Coup



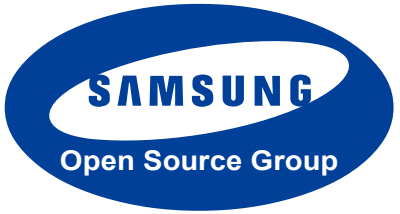
- **Libav** emerges
- FFmpeg wiped out from Debian / Ubuntu (Package maintainer was part of the take over group)
- Bikeshedding continues: people fighting about the logo!
- Consequences beyond that: FFmpeg out of GSoC

Coup Aftermath



- Michael's renaissance
- Michael starts merging back Libav improvements. Trying to maintain compatibility between the two projects
  - This process will grow to become Michael's main activity latter on

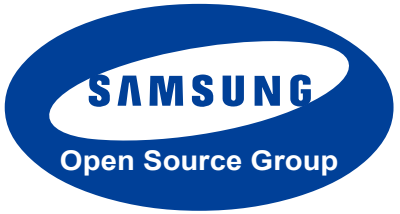
Coup Aftermath



- Libav plays by **'FFmpeg doesn't exists'** game terms. FFmpeg-incompatible changes get pushed without any cross-project coordination (Not that anyone was really expecting coordination to happen)
- New reviewing rules on Libav had an impact on its development pace (Moves slower than FFmpeg)

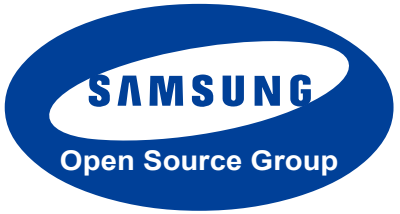
Stabilization

Stabilization



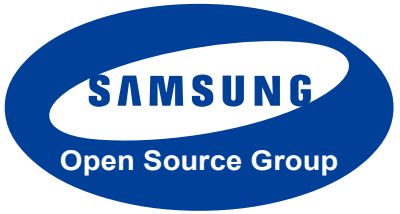
- It took a while for the teams to start thinking past the coup
- For many it was quite a traumatic event
- For the FFmpeg-relying community it was certainly a loss

Stabilization



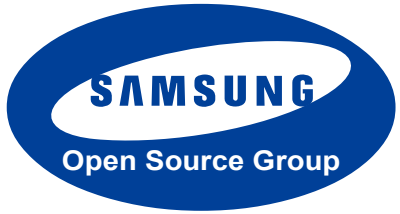
- FFmpeg development pace keeps thriving. Some examples:
  - HEVC ratified as a standard April 2013, committed Oct 12 same year
  - Native VP9 decoder, finalized June 2013, committed Sep 2013

Scenario in 2014



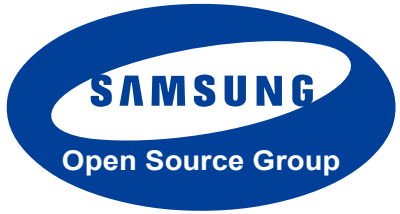
- Single lead, managed community (Michael Niedermayer since 2004)
- File-level maintainership model still in place
- Project lead only resolves on conflict
- Informally appointed community-leads handle interaction with the rest of the world. Michael is active and oversees these aspects too

Scenario in 2014



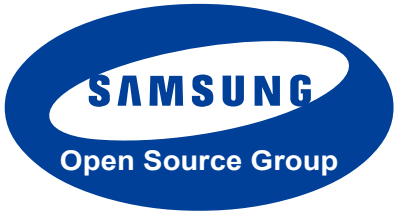
By 2014, most of what the revolution had ruined had been fixed. Pretty much everything but the general feeling of unrest and the split community.

Scenario in 2014



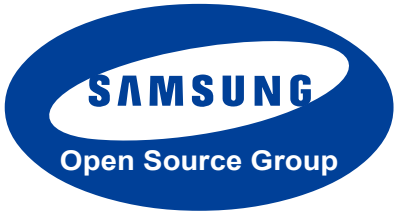
- FFmpeg back to GSoC
- FFmpeg on OPW / Outreachy
- FFmpeg back to Debian (Oct 2014)
- FFmpeg as libav provider on GStreamer 1.6 (2015)

But Things Were About to Change



yet another time...

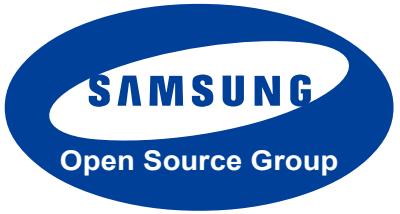
Today



- ***Michael submitted his resignation as project lead***
- A community of voters was selected to make decisions on conflict
- Initial voter committee was selected purely on development stats

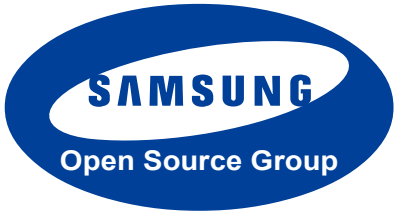
Current Challenges

Current Challenges



- As for any other FOSS project, the main external issue is staying relevant ***per se***
- The main internal one for FFmpeg: Staying together

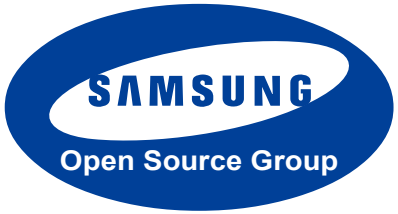
Current Challenges



FFmpeg faces challenges on several areas

- With its users
- With the industry
- With its internal organization

FFmpeg and Its Users

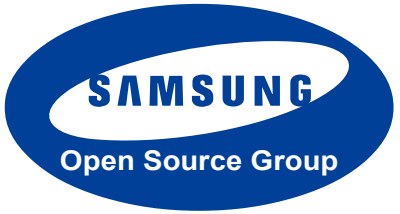


Since it is both a command line tool and a library, its users can be categorized depending on their usage approach

- Users
- Developers

It is mportant to mention that the above doesn't always hold true (Like most things in life)

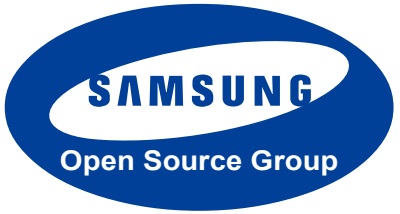
FFmpeg and Its Users



Users

- Shrinking user base
- Field progress and HW development makes tweaking largely irrelevant up to an acceptable quality baseline
- Basically no way around it but focusing on the library side (This is something Libav did seemingly right away)

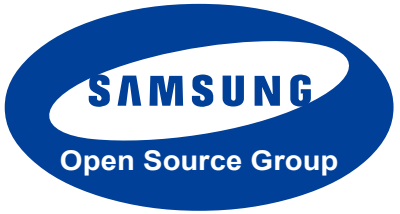
FFmpeg and Its Users



Developers

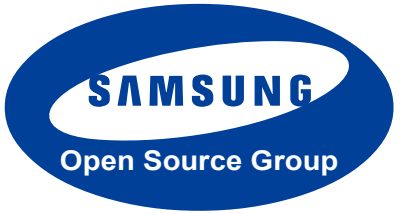
- User base shifting (?) towards service developers
  - Light and low level library doesn't save you much on complex processing use cases but perfectly fits the needs of largely-distributed and scalable single-purpose systems
- Most of the challenges here intersect the ***industry challenges***

FFmpeg and the Industry



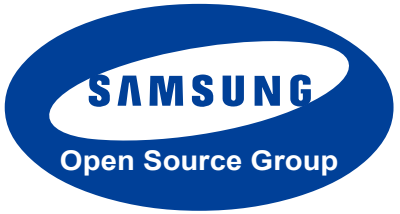
- Patents have always been an issue with FFmpeg, specially for industries that operate globally
- Huge set of corporate users but few come forward to say "I do"
- This has an immediate consequence:
  - Corporate support is basically non-existent

FFmpeg and the Industry



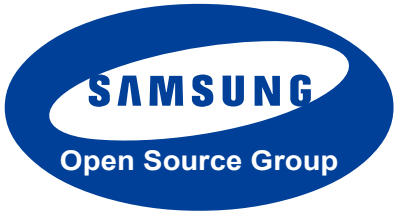
- Project management style & the lack of a formal release cycle hindered the project / industry relationship right from the start
- But there was no alternative around so FFmpeg started been **covertly** used quite fast
  - Projects like GStreamer were not an alternative, FFmpeg solves a completely different issue. These are complementary pieces of the multimedia processing puzzle

FFmpeg and the Industry



- Gained importance while hunting for **the Linux desktop**
- Circa 2008, user awareness starts to vanish. Platform switch to the smartphone starts making SW codecs less relevant (More constrained resource set, less power consumption allowance, etc)
- Continued importance on server side, non real-time processing
  - **Real life big-company example removed**
  - **Real life big-company example removed**

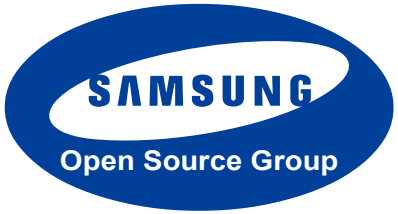
FFmpeg and Its Team



- Organize around an stable government system
- Implement **democratic** control without sacrificing development speed - agility
  - Libav is a good example. The ***FFmpeg back to Debian*** campaign had a major argument around security-bugfixing speed
- Re-unite with Libav?
  - This movement might now be accelerated by a group of developers pushing towards reunion

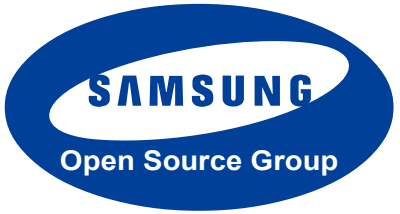
Lessons Learned

Lessons Learned



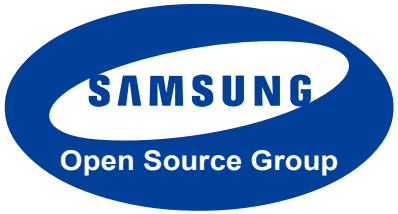
Chance of experience the extremes as a developer for both FFmpeg and GStreamer (and some other projects in between) has taught me a few things on approaching and surviving a harsh development community.

Approaching



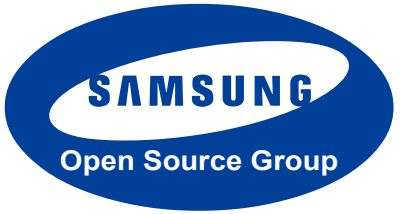
- Know: Low level software communities are ***most likely*** going to be harsh
- Accept: ***Growing a thicker skin*** is the ticket price for the fun ride
  - *TS* is pretty much the dressing code for this party

Approaching



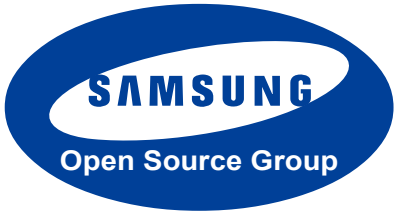
- Avoid trying to make your code perfect before putting it up for review
  - Not worthy
  - It is ***not perfect***
- Avoid proposing new things — Concentrate on fixing

Surviving



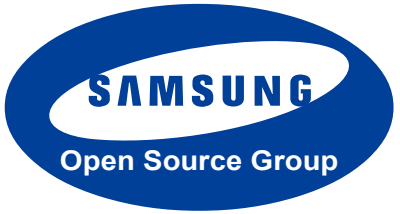
- Ignore hostility or you will get beaten up (Don't underestimate trolling and bullying experience)
- Be social, be there

Surviving



- Remember: Likely not a democratic setup, with a high tendency for bikeshedding
  - Avoid feeding locked-up-at-not-agreement discussions
  - There's an obvious, underlaying reason for this. People need the scape it provides
- Assume responsibility: aim at maintaining if you have the time

Surviving

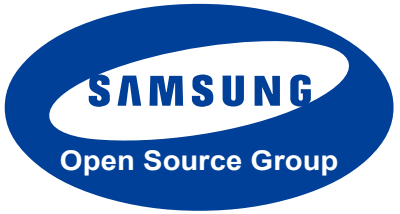


Above all: Enjoy it

SAMSUNG
Open Source Group

Enjoy It!

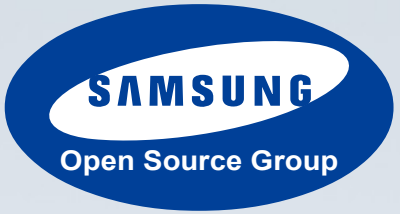
Credits



reynaldo@osg.samsung.com
@Reyverdejo

"'The Coup" image from WikiMedia Commons. Originally by Denoel Paris and other photographers - "1989 Libertate Roumanie" by Denoel Paris (ISBN 2-207-23695-1)"

Questions



Thank you.