



**GStreamer**  
***Past - Present - Future***

Wim Taymans ([wim.taymans@gmail.com](mailto:wim.taymans@gmail.com))

*26 oct 2010 - GStreamer Conference  
Cambridge, UK*



# Overview

## gstreamer tools

gst-inspect  
gst-launch  
gst-editor

media player

VoIP & video  
conferencing

streaming  
server

video editor

(...)

## multimedia applications

### gstreamer core framework

#### pipeline architecture



media agnostic  
base classes  
message bus  
media type negotiation  
plugin system  
utility libraries  
language bindings

#### protocols

- file:
- http:
- rtsp:
- ...

#### sources

- alsa
- v4l2
- tcp/udp
- ...

#### formats

- avi
- mp4
- ogg
- ...

#### codecs

- mp3
- mpeg4
- vorbis
- ...

#### filters

- converters
- mixers
- effects
- ...

#### sinks

- alsa
- xvideo
- tcp/udp
- ...

## gstreamer plugins

gstreamer includes over 150 plugins

**3rd party  
plugins**

## Past

- Started in early 1999 by Eric Walthinsen
  - Based on OGI reseach
  - Ideas from DirectShow
  - C, G(tk)Object
- State of multimedia on linux was very poor back then
  - Xanim<sub>+</sub> (vlc 1999, Ogle<sub>+</sub> 1999, xine<sub>+</sub> 2000, mplayer 2000, ffmpeg 2000, ...)
  - No codec libraries
- Make something that can compete with Quicktime (1991) /DirectShow<sub>+</sub>(1996)

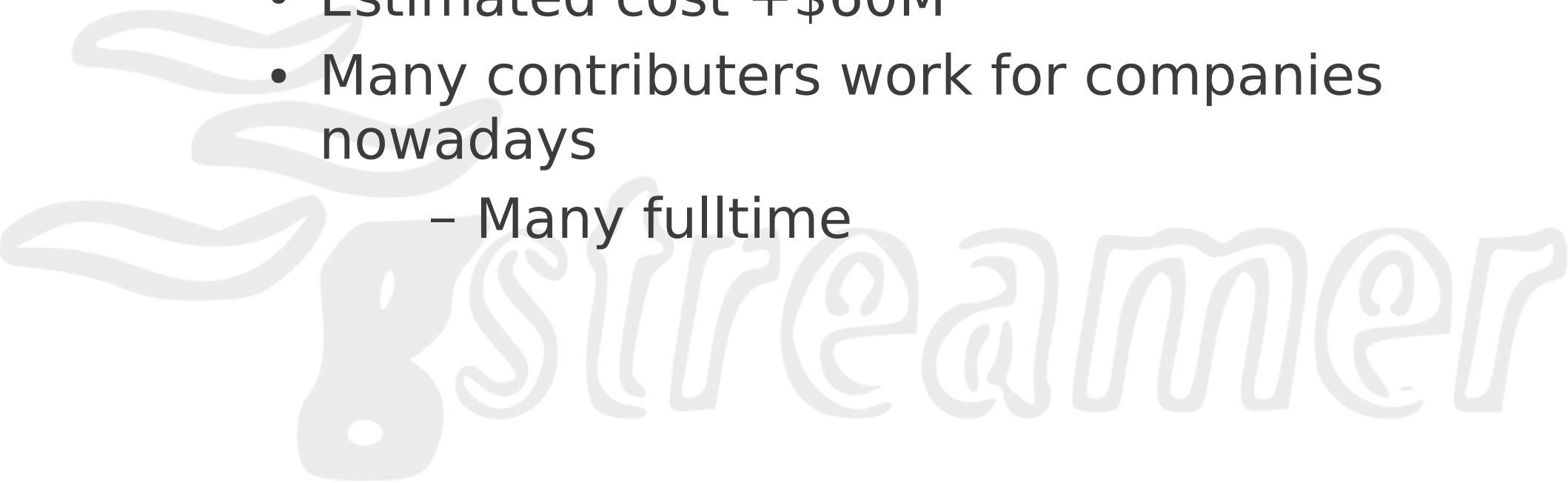
## Use cases

- Music players (Rhythmbox, Songbird,...)
- Video players (Totem, Webkit,...)
- Streaming servers (Axis, UbiCast, Flumotion, RTSP, DLNA server...)
- Transcoding (Arista, Transmageddon, ...)
- Media capture (Cheese, N900, ...)
- Audio editing (Jokosher, ..)
- Video editing (PiTiVi, ...)
- VoIP (Empathy, Tandberg, ...)
- Desktop, embedded

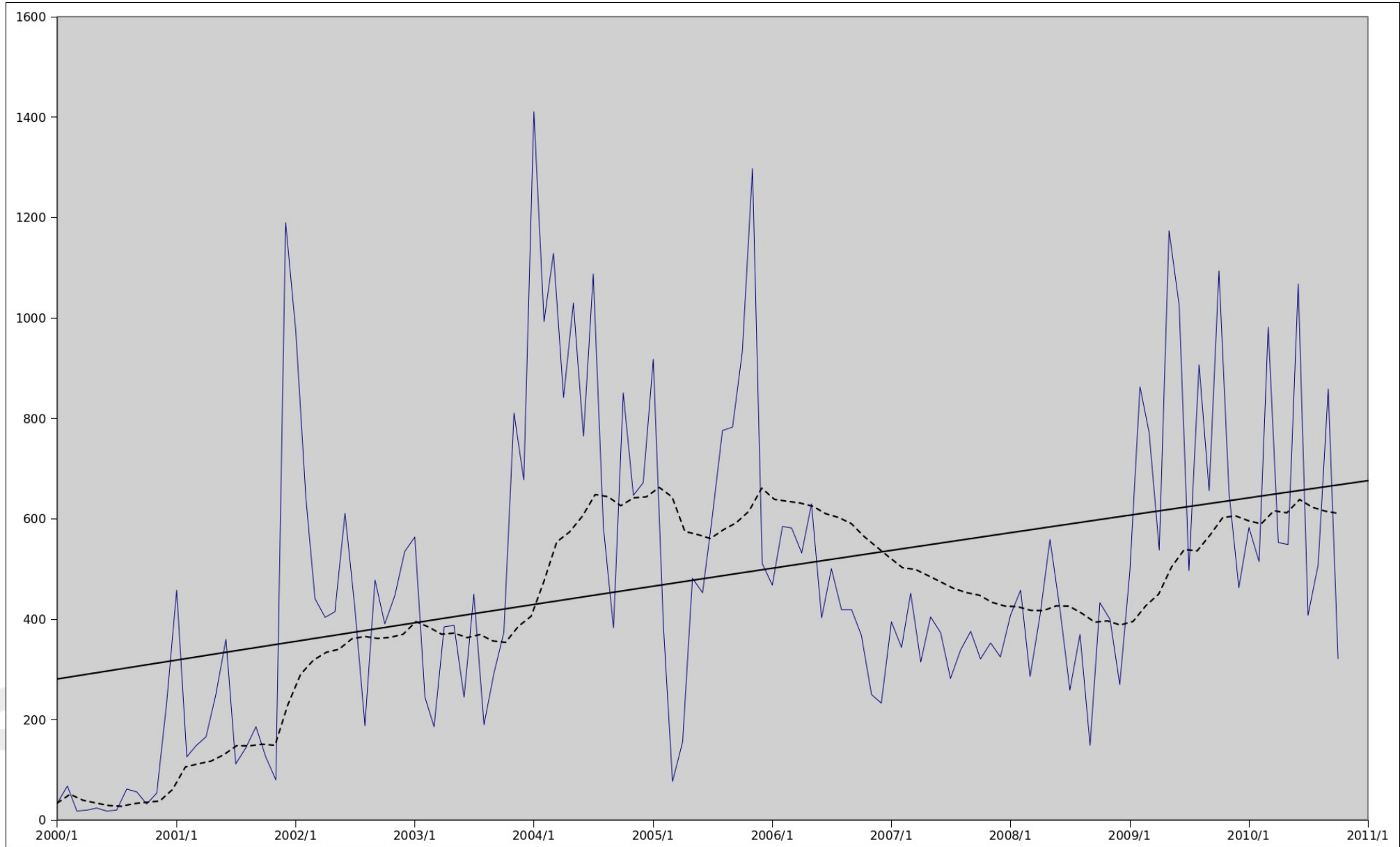
## History

- 0.0.1 – Jun 1999 (POC)
- 0.1.0 – Jan 2001
- 0.2.0 – Jun 2001
- 0.3.0 – Dec 2001
- 0.4.0 – Jul 2002 (in gnome, very rough)
- 0.6.0 – Feb 2003 (audio works ok)
- 0.8.0 – Mar 2004 (video works okish)
- 0.10.0 – Dec 2005 (redesign, maturity)
  - Currently 0.10.30
- 1.0 - ???

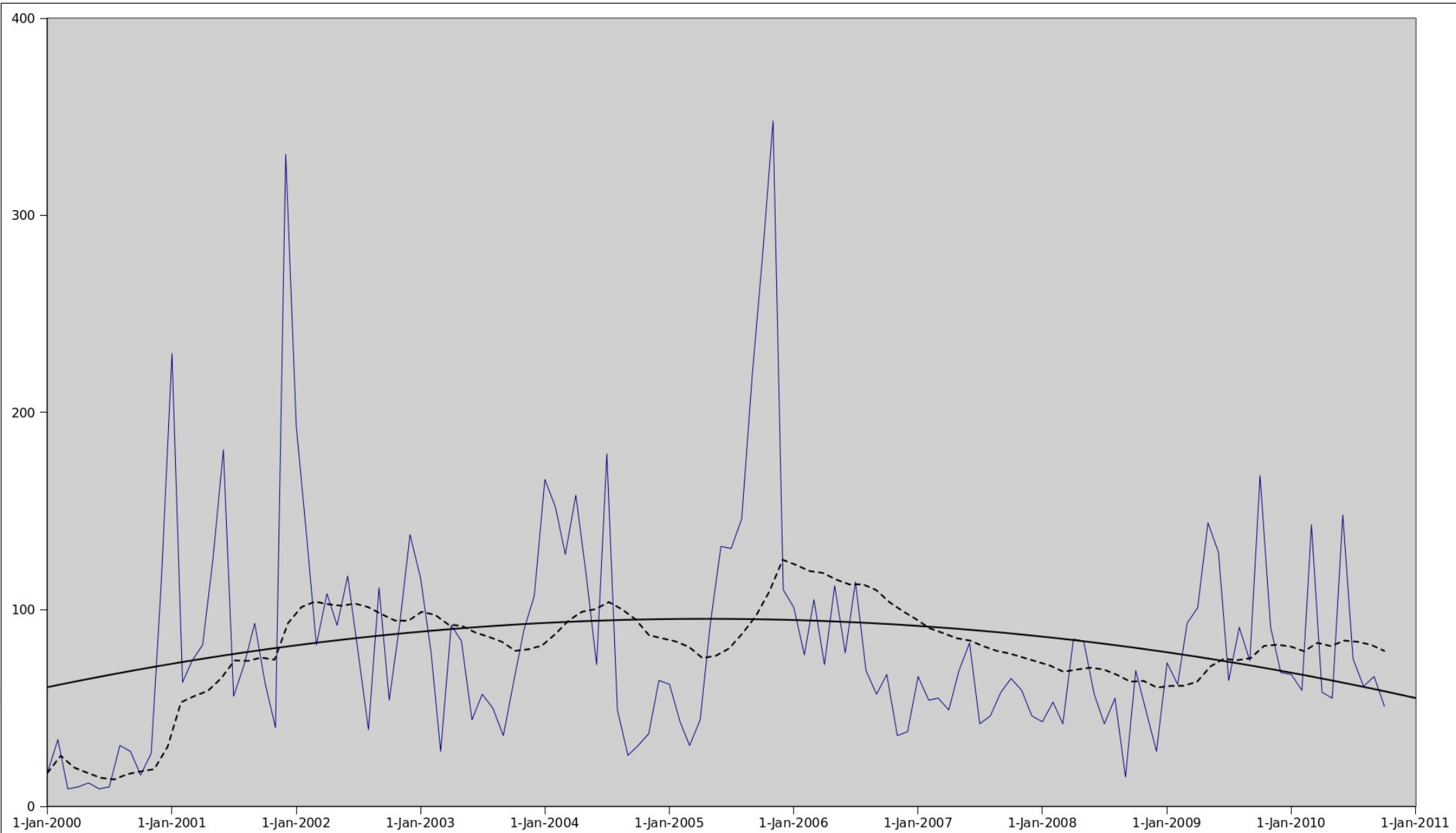
- large community
  - 30+ core developers
  - 200+ contributors
- Core 205K LOC
- Plugins +1M LOC
- Estimated cost +\$60M
- Many contributors work for companies nowadays
  - Many fulltime



# Overall commits per month

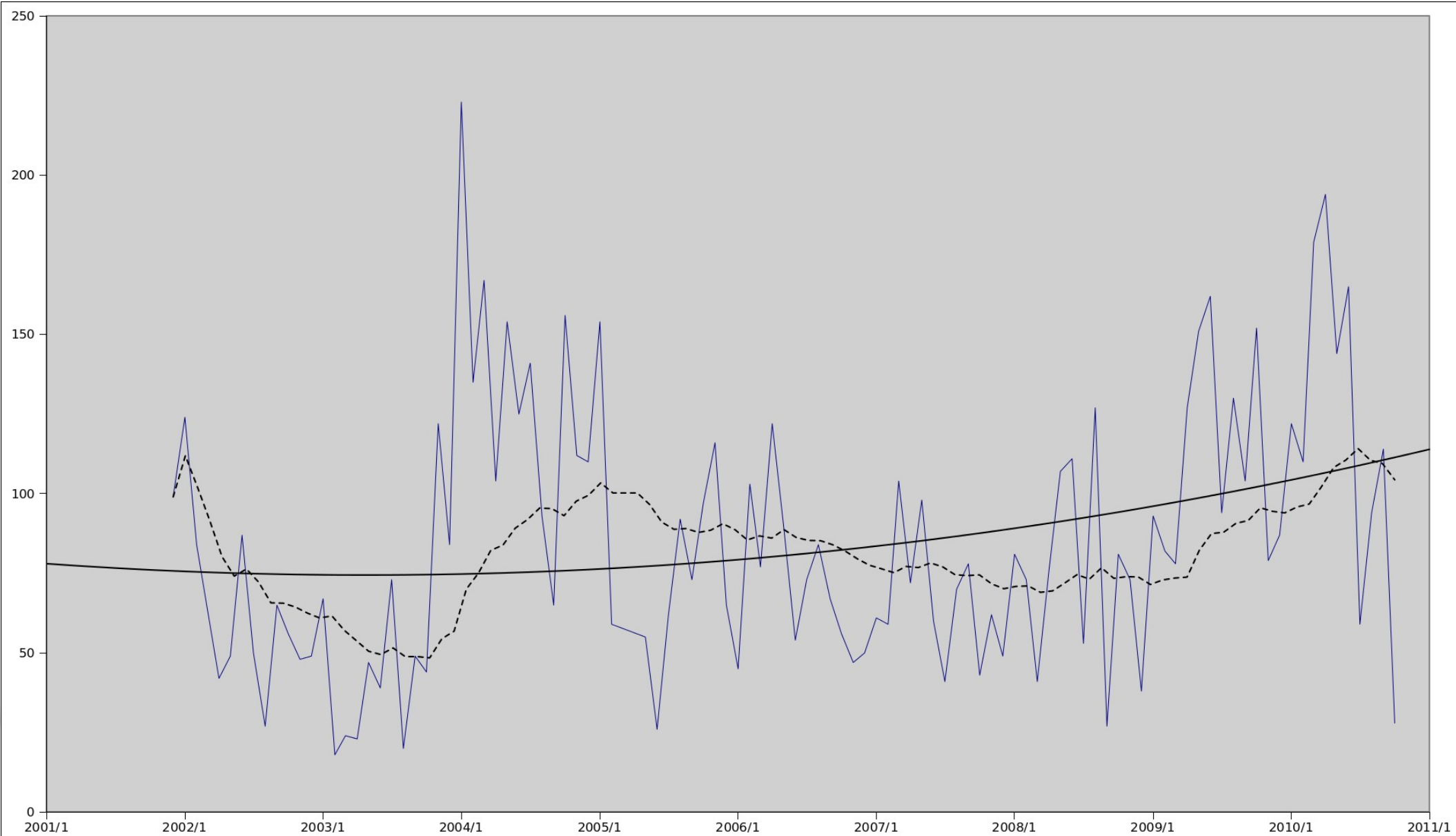


# Core commits per month

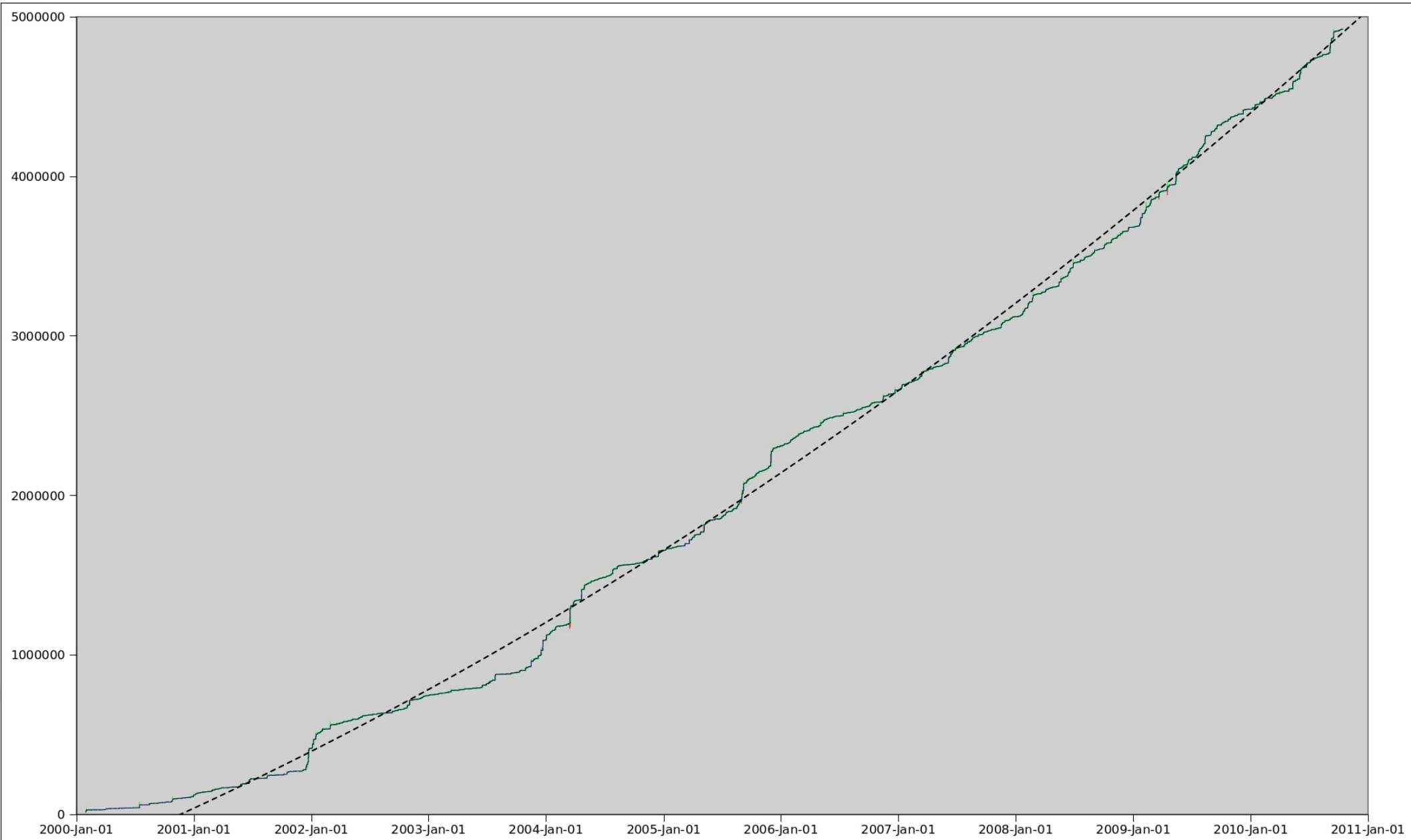




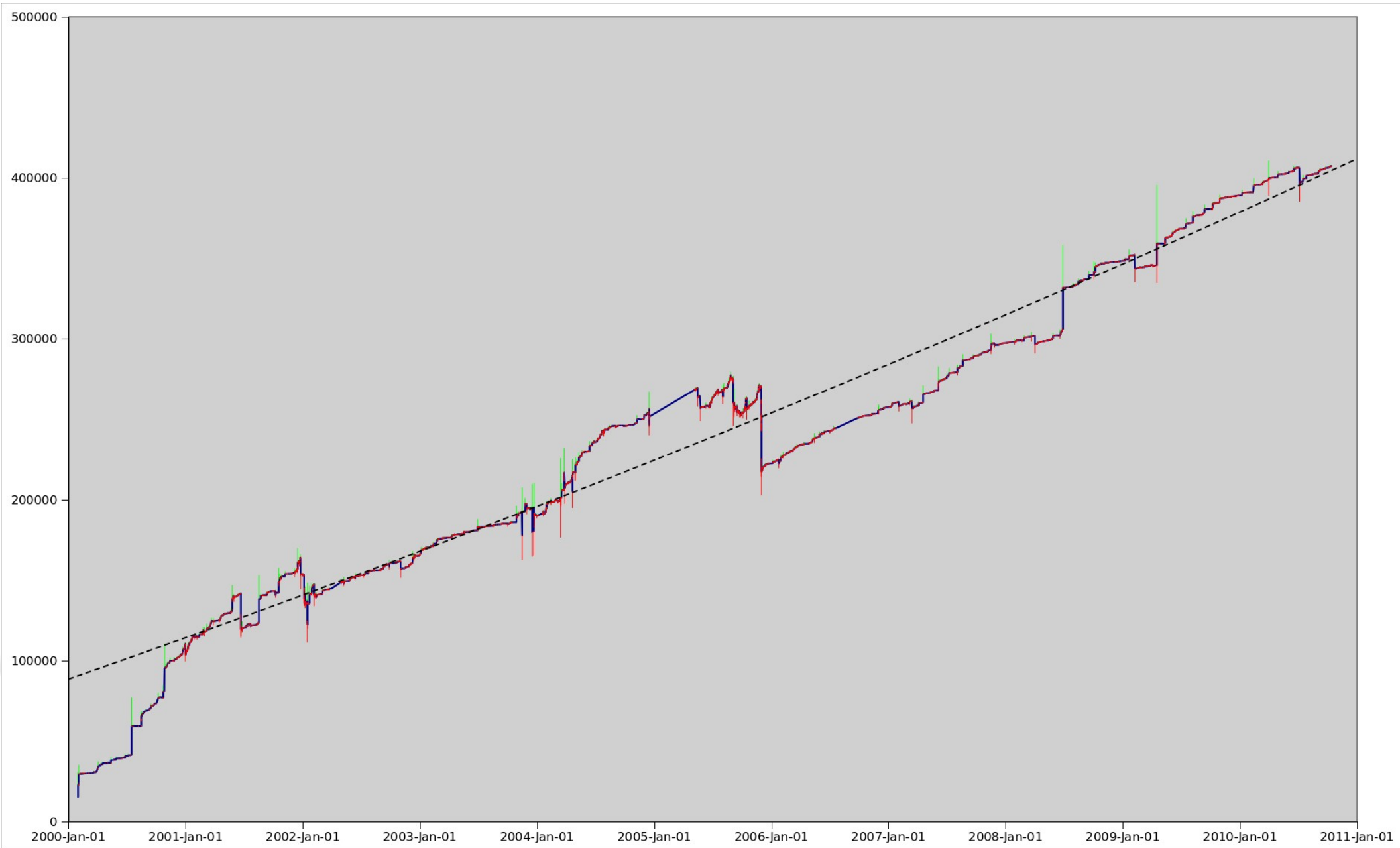
# plugins-good commits per month



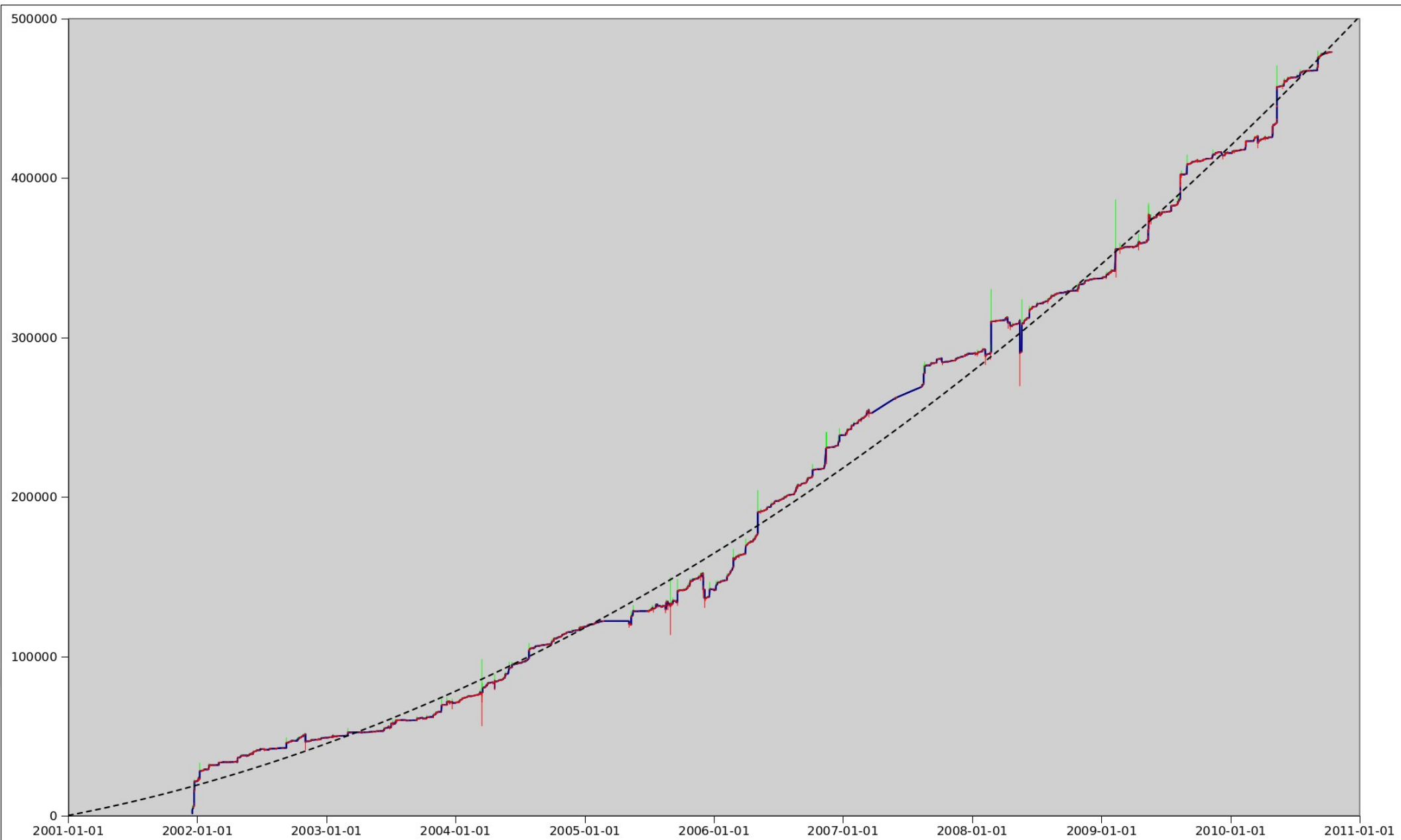
# overall lines of code



# Core lines of code



# Plugins-good lines of code



## Some Features

- Fully multithreaded
- Advanced trick mode playback
- Video editing support
- Top notch RTP/streaming support
- Extensive, well tested, format support
- Accurate robust synchronisation
- Advanced autoplugging, dynamic pipeline changes
- Live sources, clock slaving, low-latency,...
- Binding friendly (python, C#, C++, vala, perl, scheme, ruby, java, ...)
- Extensive debug system

## 0.10 good points

- Extensibility of the 0.10 design
  - Padding to all public structures
  - Addition of private structures
  - New events/queries/messages/caps without API/ABI breaking
  - Rewrote various parts like state changes, clocks
  - Implemented navigation, QoS, stepping, latency, stream-status, buffering, trickmodes, ...

- Higher level objects
  - Playbin2
  - Encodebin
  - Tagreadbin
  - Farstream
- Base classes
  - Sink/source/transform
  - Decoder/encoder
  - Parser
- Helper libraries

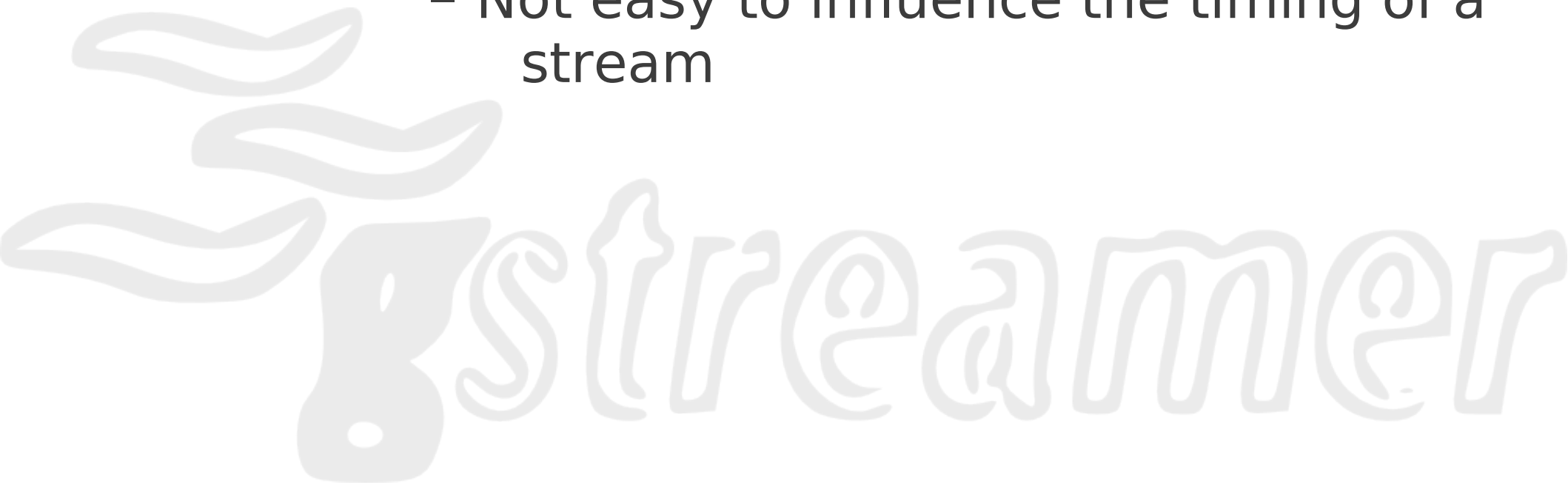
- Negotiation
  - `gst_pad_get_caps()` can't be optimized for speed
  - Rethink reverse negotiation (probably with an event)
- Caps too verbose
  - `video/x-raw-rgb, bpp=16, depth=15, endianness=1234, red_mask=31744, green_mask=992, blue_mask=31`  
=> `video/x-raw, format=RGB15`



## 0.10 bad points

- No extensible buffer metadata
  - GstMiniObject subclasses too limited
  - We can't express strides, per plane pointers, ..
- Need generic ways to map buffer data
  - For DSP, GPU
  - For doing cache flushes
  - Abstract other image APIs (cairo, opengl, ...)

- Dynamic pipeline modifications
  - Not easy with newsegment events
  - Not easy to influence the timing of a stream



## 0.10 bad points

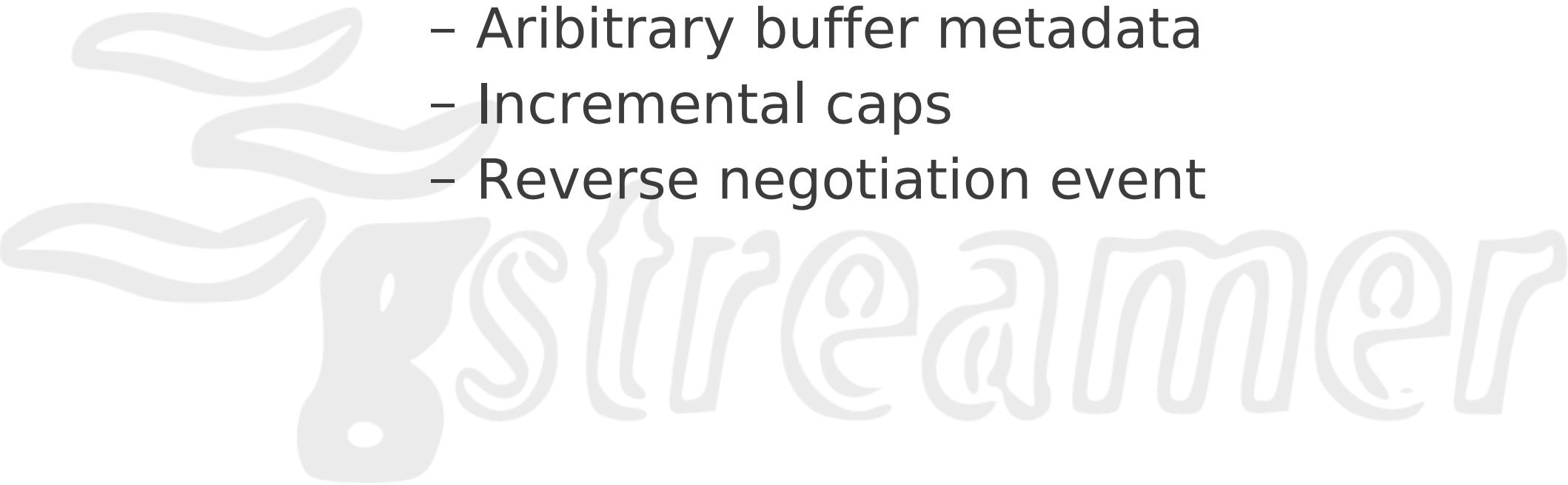
- We collected a fair amount of deprecated API
- We're out of padding
- Some new features hard to implement without breaking API/ABI
- Some APIs just needs changing to move forwards

## Short term plans

- Continue improving plugins
  - Fix bugs
  - Implement new features
- Core speedup improvements
  - Making shared datastructures lockfree (clocks, bus, queue, ...)
  - Reduce overhead in common cases (datapassing, base classes, ...)
- Improve highlevel objects
- Make more base classes, improve base classes

## medium term plans

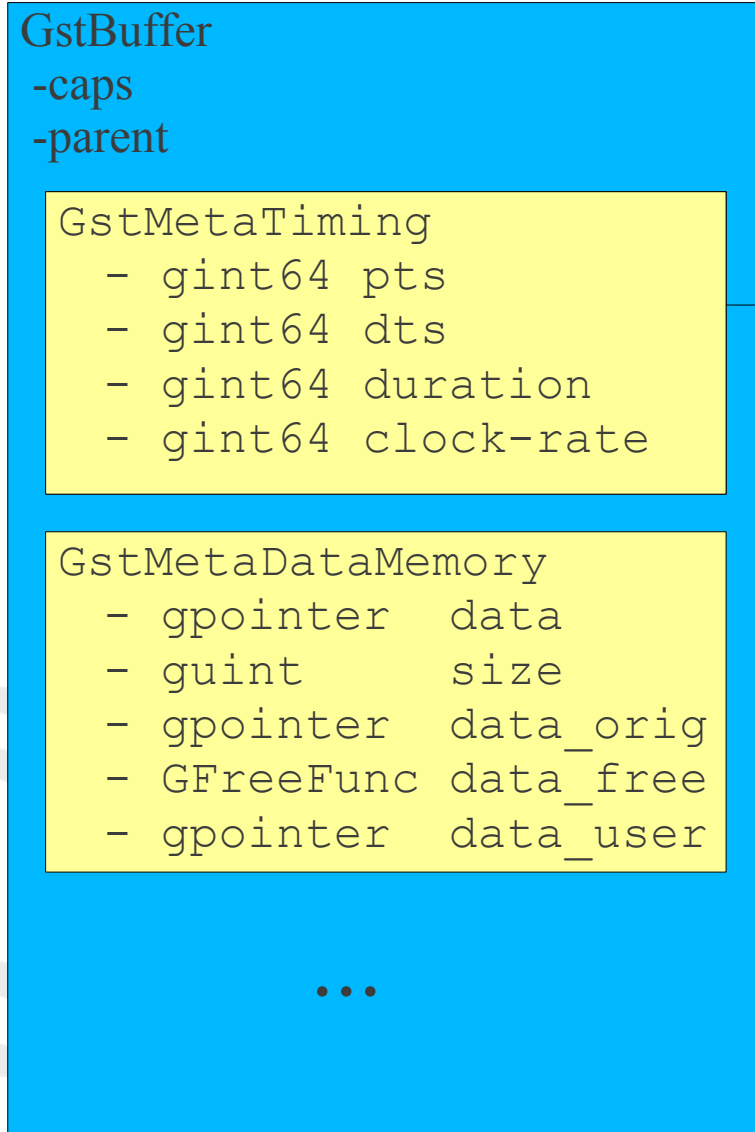
- Collect requirements
- Flesh out new design ideas
- Experiment with new things in git branches
  - Arbitrary buffer metadata
  - Incremental caps
  - Reverse negotiation event



## Buffer metadata

- Simplify GstBuffer
  - Only 2 fields : caps and parent
  - Free space for metadata
- Metadata are registered named structures
- API to add/iterate/remove metadata from buffers

# Buffer metadata



## Example : Video Metadata

- GstMetaVideoMemory
  - Pointers to planes
  - Per plane stride
- GstMetaVideoRectangle
  - Region of interest
  - Crop/zoom/pan
- GstMetaVideoCairo
  - Pointer to cairo data

- ...



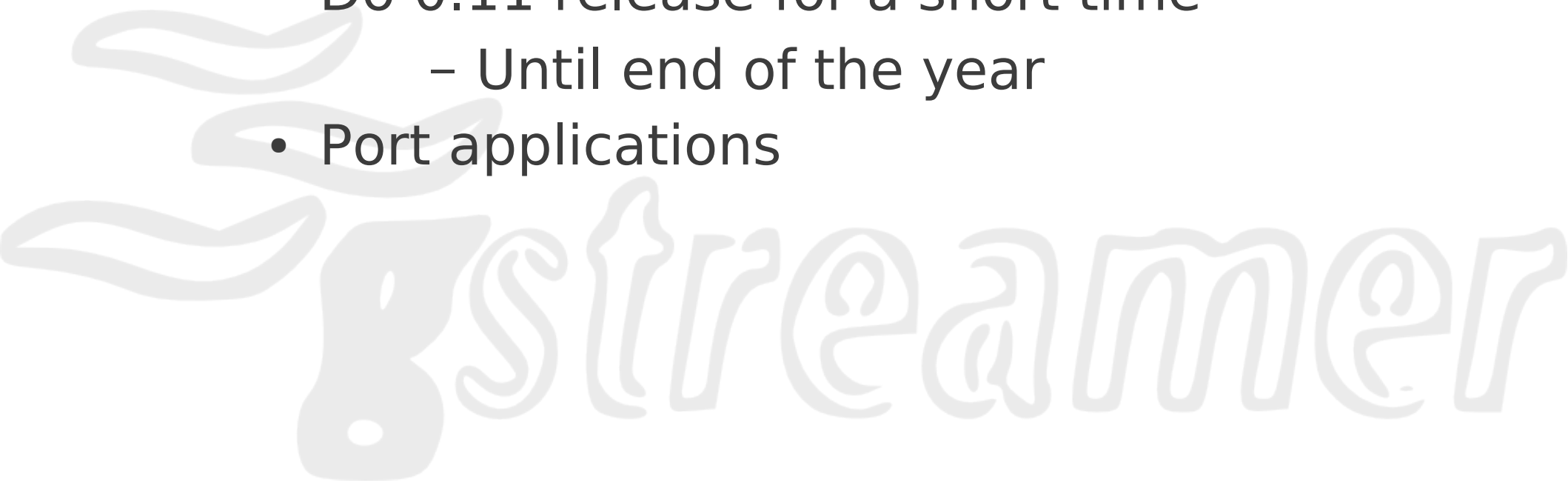
- Required metadata structures negotiated with caps ?
- Rethink `gst_pad_alloc_buffer()`
  - Prototype based?
  - Use caps to describe accepted metadata
- All elements need updating
- Avoid explosion of Metadata
  - Make it extensible
- Relation with caps ?

## Timing

- Tweak GstSegment to include the accumulated time (offset)
  - No more segment accumulation
  - Segment accumulation only useful for looping
- Make GstSegment event sticky to pads
  - Much like caps
  - Can ask running-time on pads
- Add API to change offset on pads
  - Can adjust running-time on a per pad basis

## long term plans

- Merge code into 0.11 branch
  - Starting from januari 2011
- Port all plugins
- Do 0.11 release for a short time
  - Until end of the year
- Port applications



## long term plans

- Do 1.0 release !
  - Near end of 2011





World domination !





Questions ?

