Achieving Pexcellence

Håvard Graff - havard@pexip.com

Achieving Pexcellence

Challenges of real time streaming applications

Background

- TANDBERG
  - Movi
- CISCO
  - Cisco Jabber Video for Telepresence
- Video Conferencing SoftClient
  - Think Manageable and Standards-based Skype
  - Uses GStreamer
  - Several 100K deployments

Problems

- Billions and Billions of Threads (> 1)
  - Race conditions
  - Unexpected Behavior
  - Deadlocks
  - Crashes
- Real-Time (Live) System
  - Never reproducible results
  - Experienced choppy audio?

Basic Solution

- 10 Reproduce or Induce problem with a Test
- 20 Fix it
- 30 goto 10

Basic Solution

- 10 Reproduce or Induce problem with a Test
- 20 Fix it
- 25 Commit Test and Fix into your CI
- 30 goto 10

But how?!?

- Not preaching TDD, but…
  - Writing good tests are hard, and where your focus *should* be…
  - Too much “brilliant” code has crap tests. (if it’s lucky…)
  - All code has bugs.
  - But testing will find more of them.
- Show you our approach:
  - Not perfect, but we like it (more && more)
  - Interesting to hear other approaches as well!

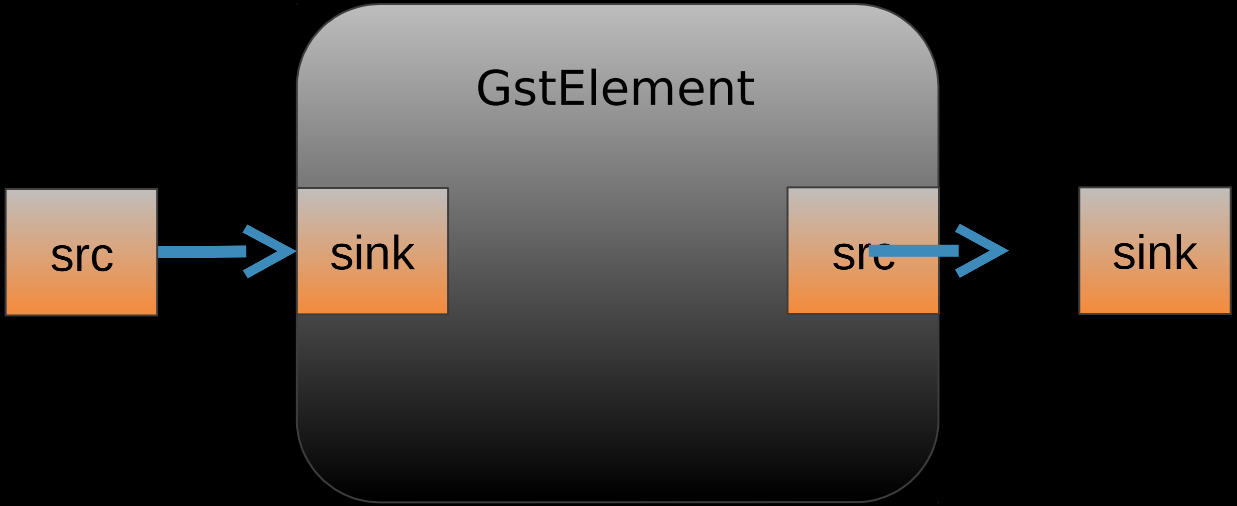
GstCheck

- Framework for Testing
- Easy to:
  - Write Tests
  - Run Tests (make mytest.check)
  - Debug Tests (make mytest.gdb)
  - Test Tests (make mytest.forever)
- Valgrind integration (make mytest.valgrind)
  - With suppression!
- Beginnings of a framework for testing GstElements

GstHarness

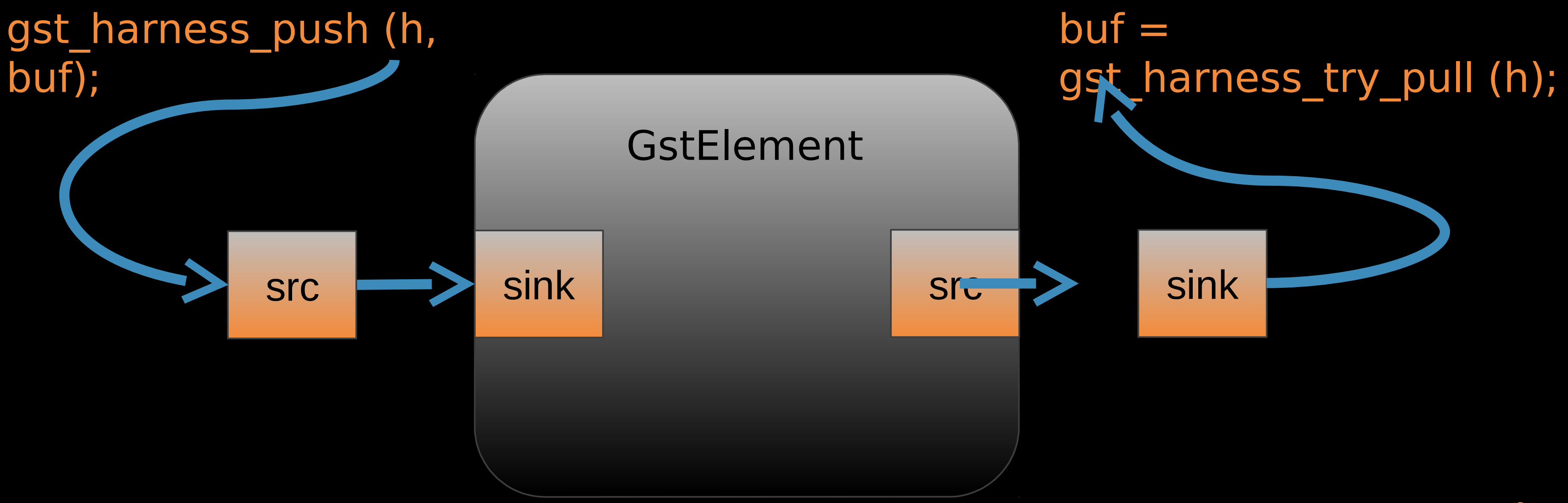
- Based on gst_check_setup_element
- Evolved on a need-to-test basis
  - Refactoring++
- Used in (almost) all our GStreamer tests (> 600)
- Goal: To easily write simple tests, testing complex scenarios!

GstHarness



pexip

GstHarness



pexip

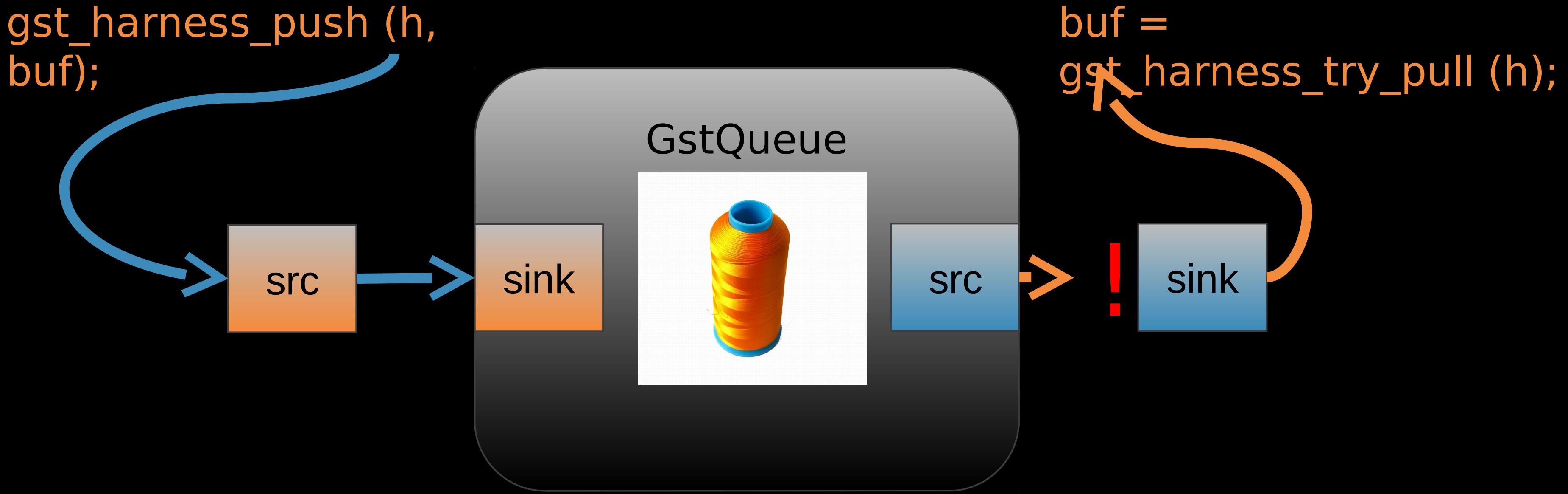
Test

Does GstIdentity modify buffers?

Test

How about a GstQueue?

GstQueue


gst_harness_push (h, buf);
buf = gst_harness_try_pull (h);
GstQueue
src
sink
src
!
sink

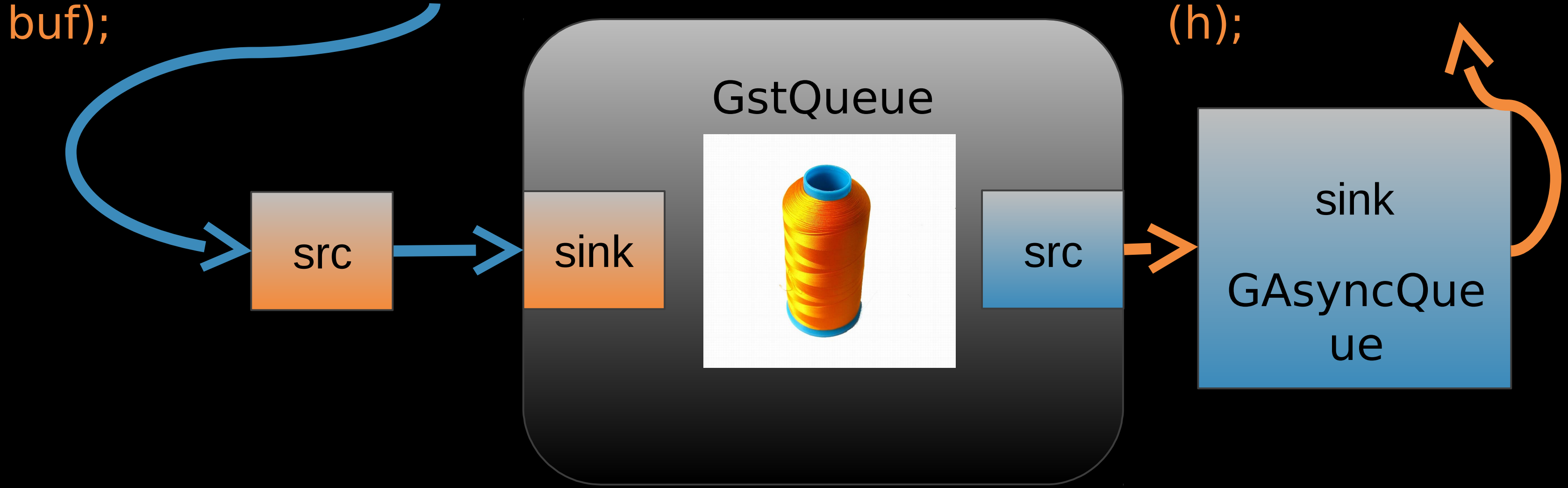
GAsyncQueue

- Perfect!
- gst_harness_try_pull: g_async_queue_try_pop
- gst_harness_pull: g_async_queue_timeout_pop
  - Remember large timeout (we use 60 sec...)
- The test finishes exactly when it should!
  - No nasty sleeps
  - You can never know how long is long enough...

Determinism

gst_harness_push (h, buf);

buf = gst_harness_pull (h);

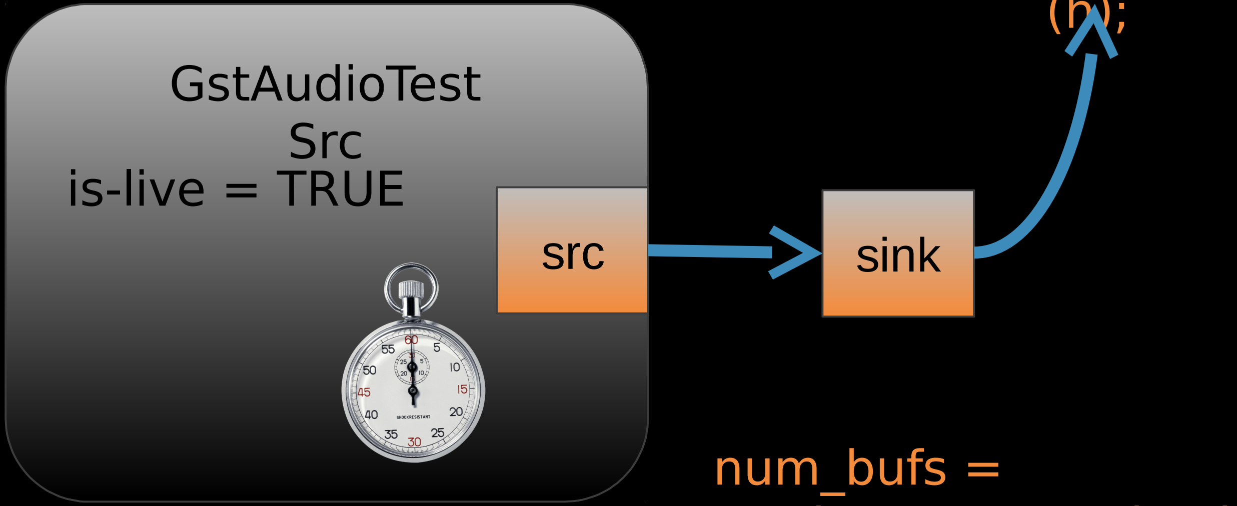


pexip

Test

Lets try a Src

audiotestsrc



buf = gst_harness_pull (h);

num_bufs = gst_harness_received_buffers (h);

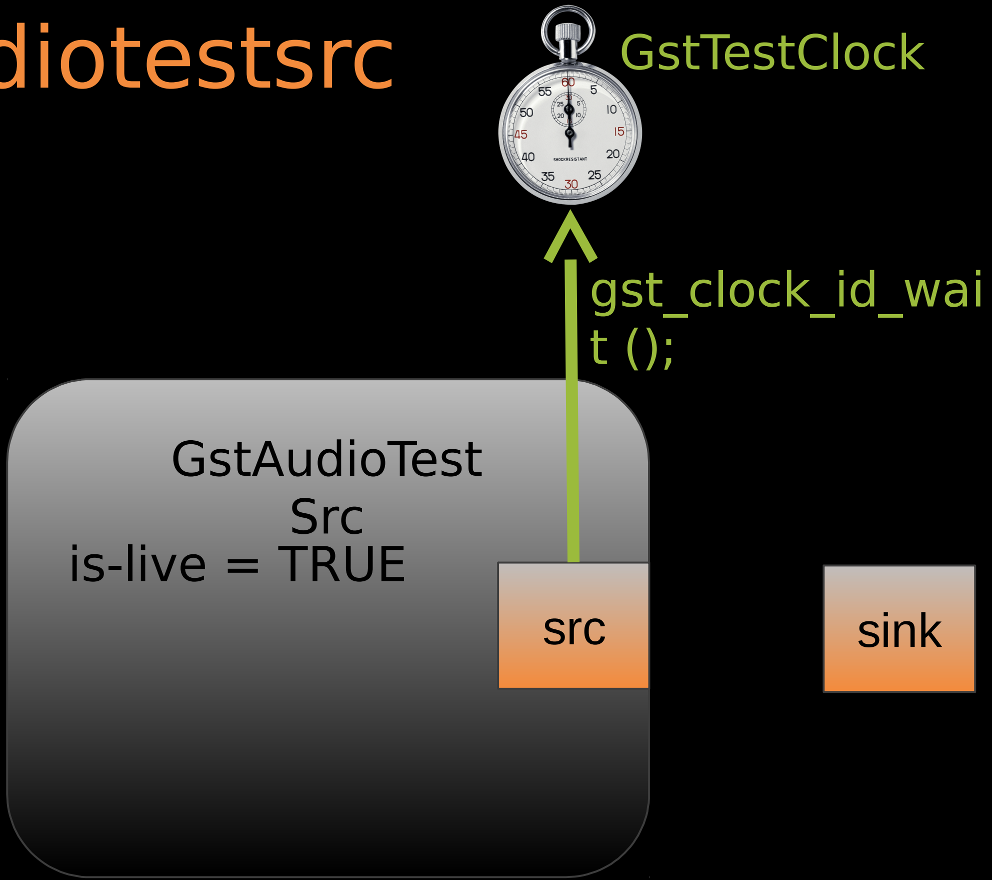
We need…

- A way to control time.

GstTestClock

- A GstClock Implementation
- Control Time
- Control GstClockID waits
- Already in GStreamer 1.0

audiotestsrc

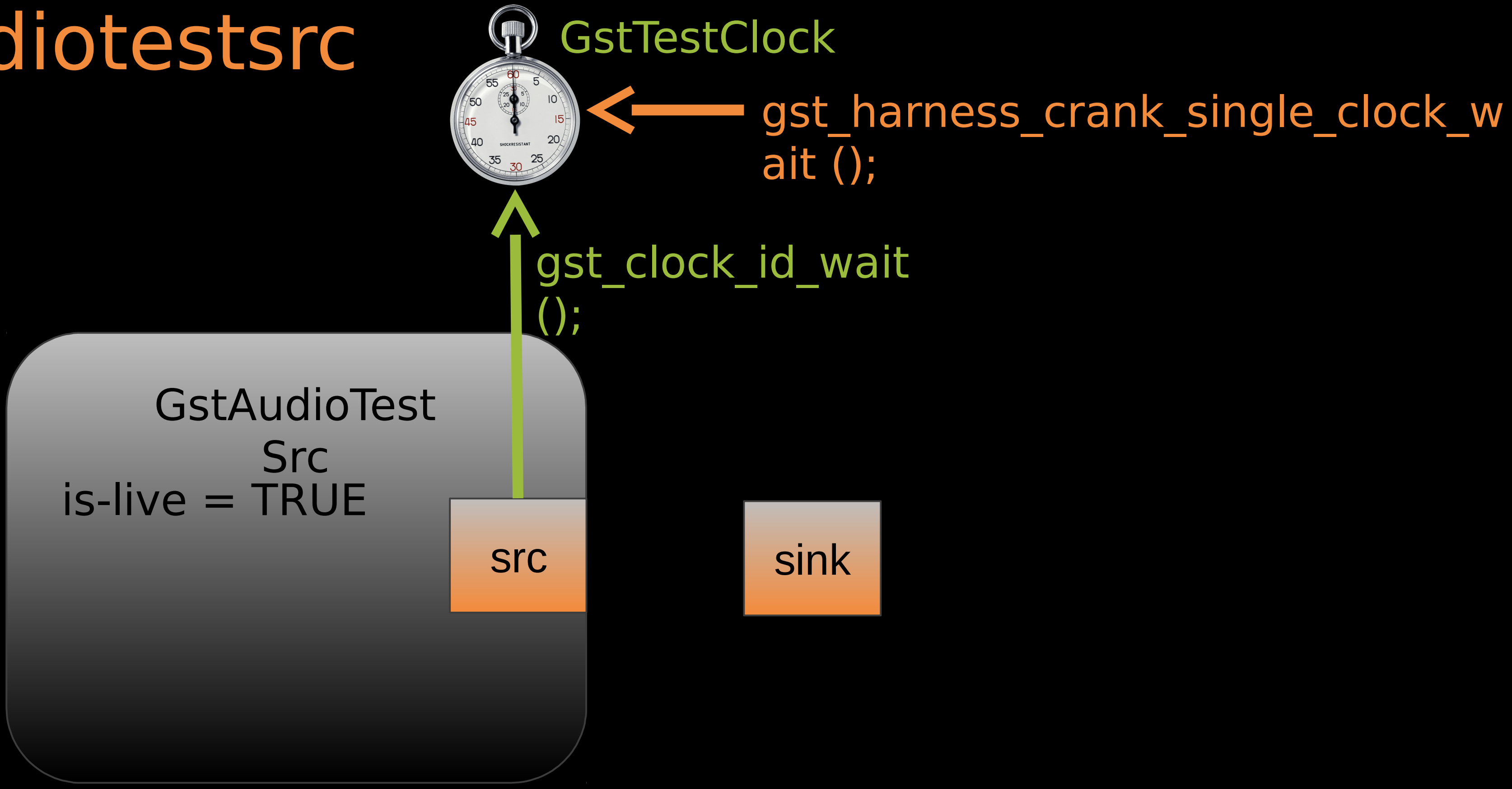


GstTestClock :: “Crank”

- 1. Wait for a given number of waits
  - Fail if not equal
- 2. Get the lowest time waited for
- 3. Advance the clock to that time
- 4. Release all waits
  - Recently added, used to be racy for >1

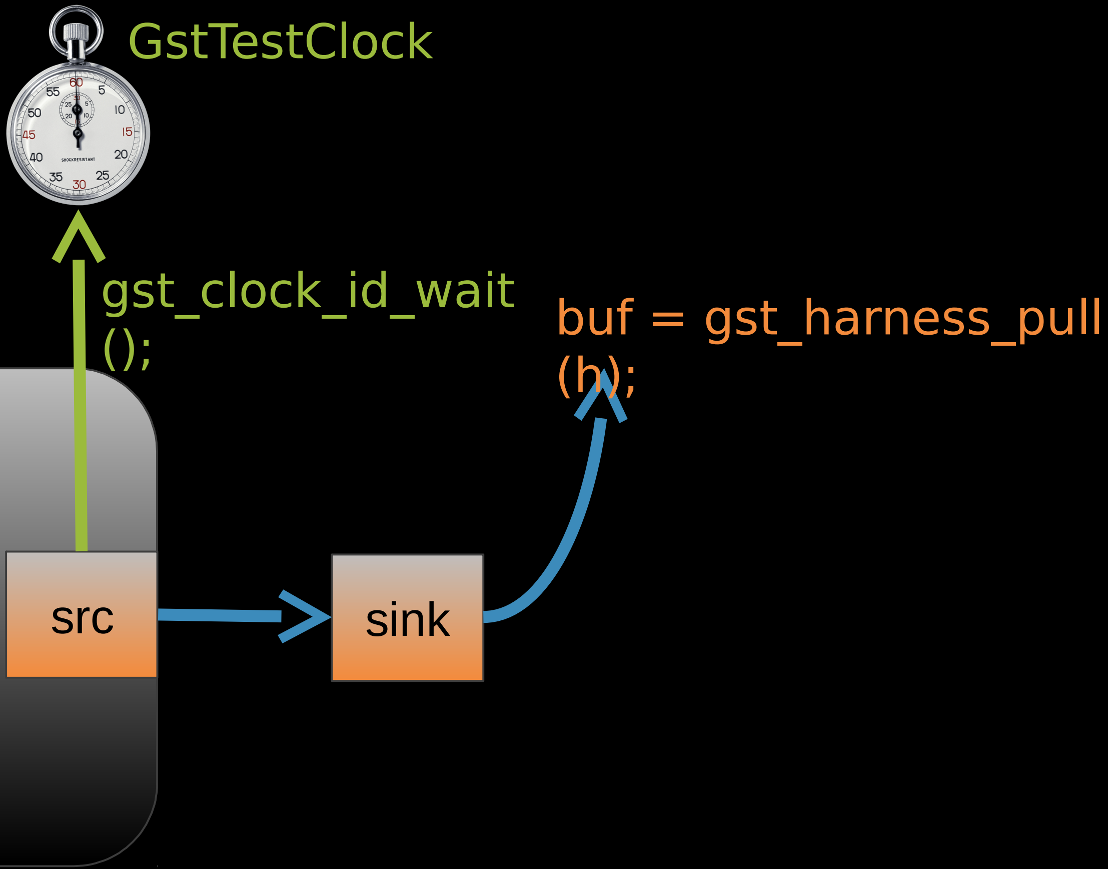
pexip

audiotestsrc



pexip

audiotestsrc



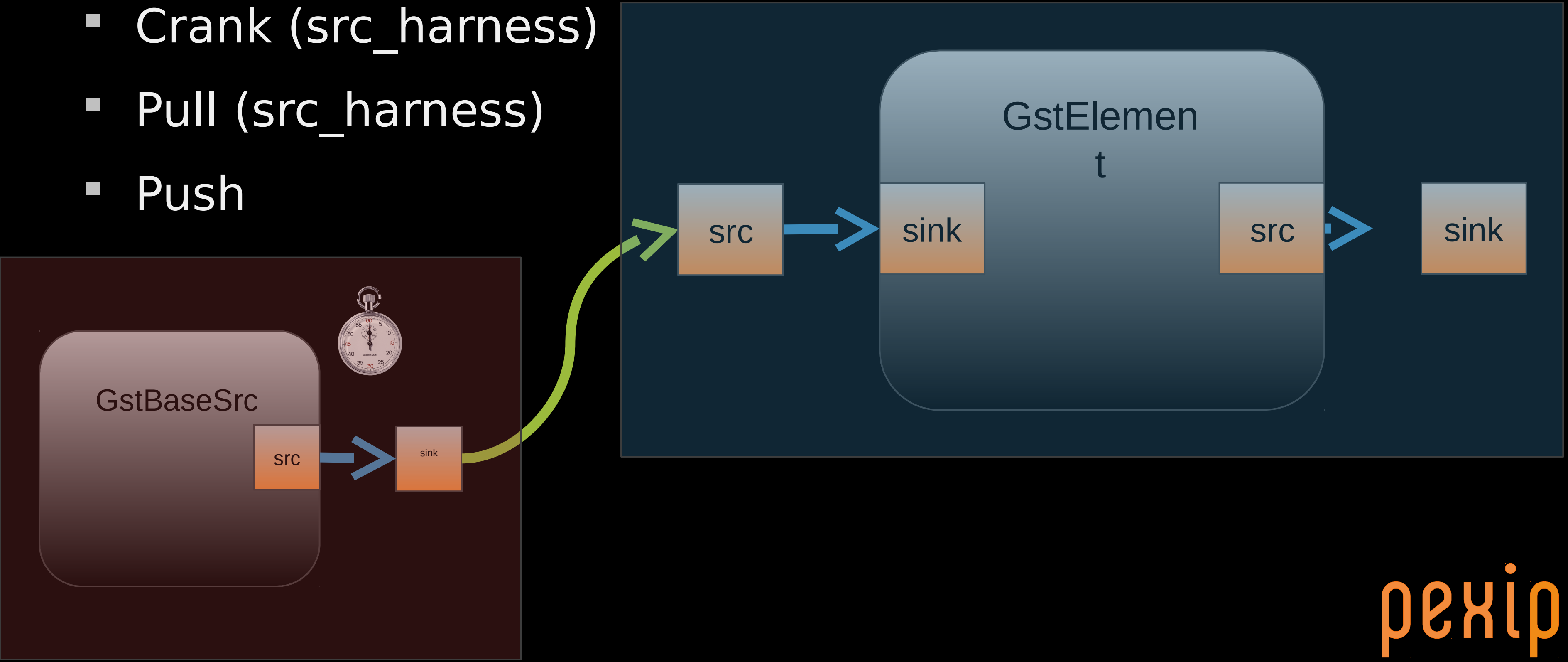
pexip

Sub-Harnesses

- Testing your element in a bigger context
- Helping keep things deterministically
- src_harness:
  - A pipeline to feed input into your element
  - Typically a src-element + friends
- sink_harness:
  - A pipeline for processing your elements output
  - Typically a sink-element + friends

gst_harness_push_from_src (h);

- Crank (src_harness)
- Pull (src_harness)
- Push



pexip

Test

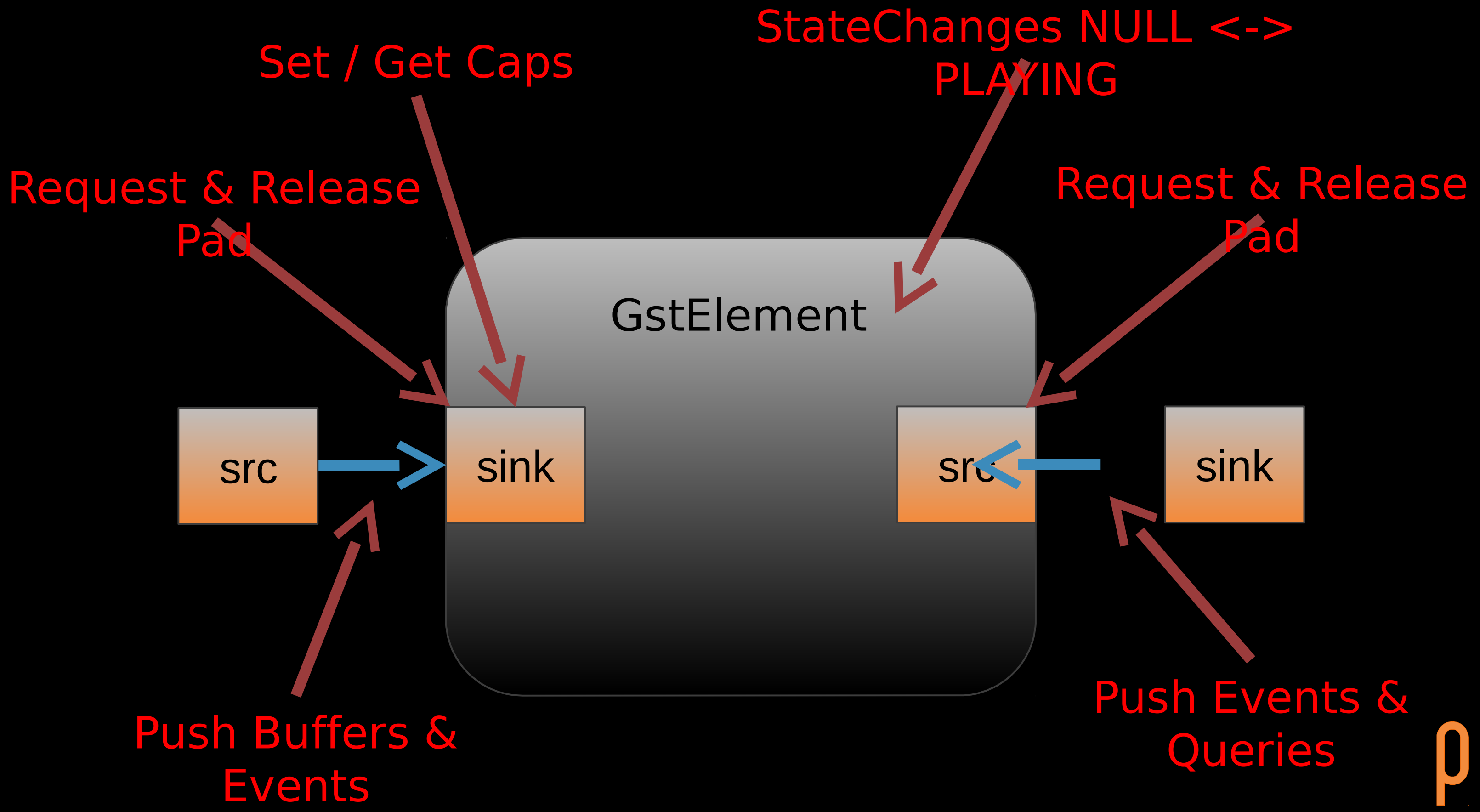
H.264 decoder sends Keyunit-Request when there is packetloss

Stress-Testing

- A complete opposite
- Very random
- Can uncover a lot of very rare crashes
- Specially powerful combined with CI
    - Some tests fail once every 2 weeks…
- Built in to the Harness

pexip

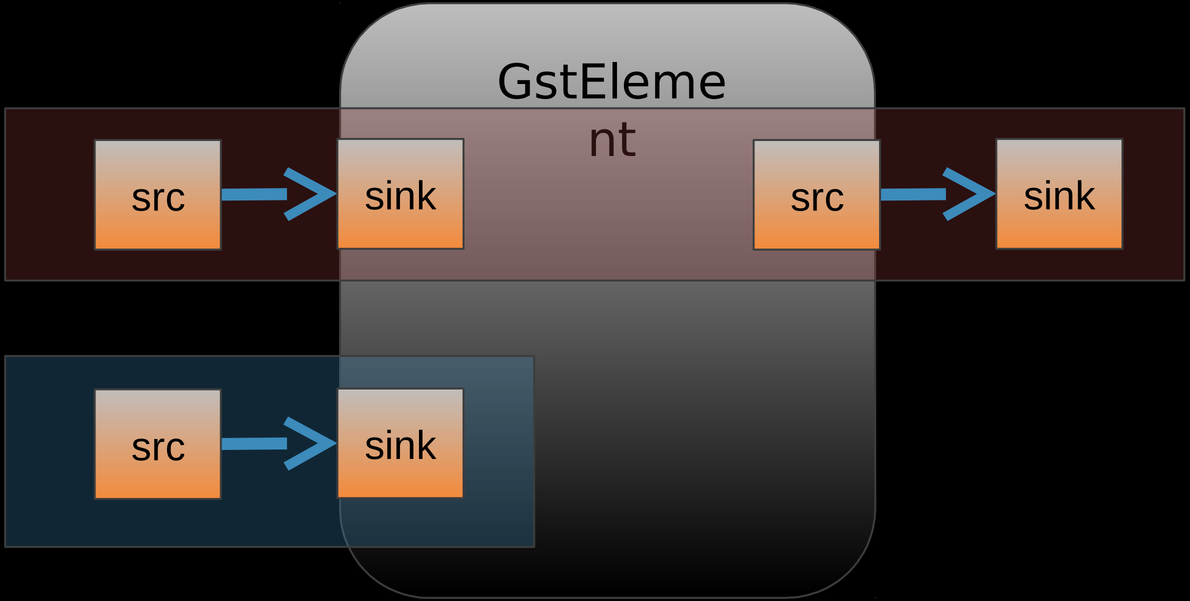
Stressing



pexip

“Multi-Harnessing”

- One harness per pad



pexip

Test

Stressing a Funnel

pexip

Further improvements

- Merging into GStreamer
  - Porting 0.10->1.0 (done!)
  - Remove Pexip-specifics
  - Make nicer / More complete
  - Start writing / rewriting tests inside GStreamer
  - Keep evolving with usecases
  - Before X-Mas! (“Ho Ho”, “From all of us” etc.)
- “GStreamer-Element Acceptance Test”
  - Do a lot of automatic checking
  - Stressing what can be stressed
  - Would catch a lot of beginner errors (and a few master ones…)

pexip

Thanks!

- Contact:
  - havard@pexip.com
  - hgr @ #gstreamer
- Questions?

pexip