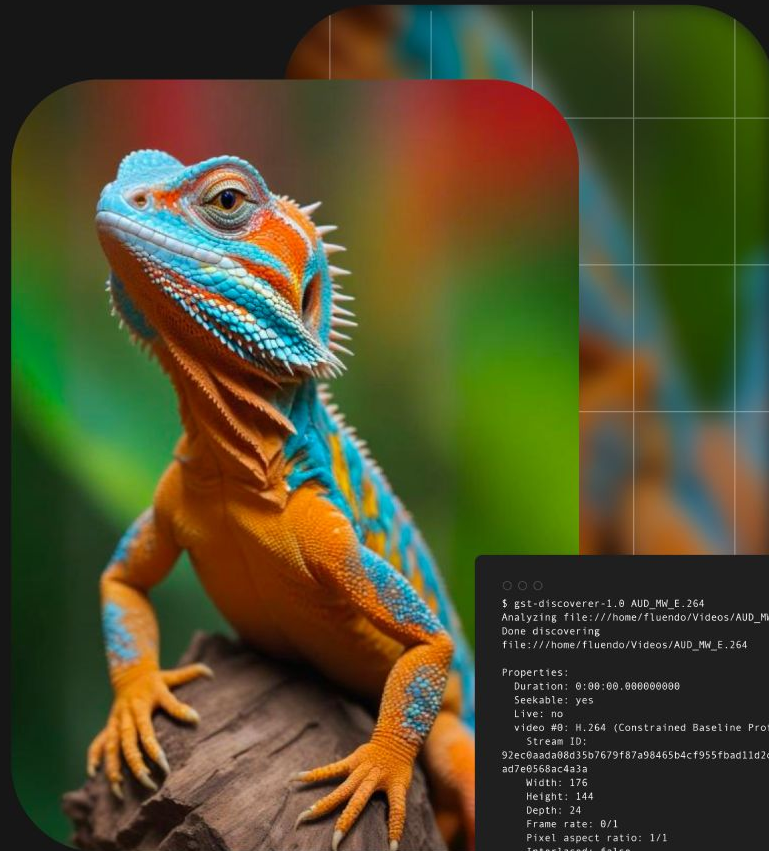




Gst Audit

Jorge "turran" Zapata
<jzapata@fluendo.com>



```
$ gst-discoverer-1.0 AUD_MW_E.264
Analyzing file:///home/fluendo/Videos/AUD_MW_E.264
Done discovering
file:///home/fluendo/Videos/AUD_MW_E.264

Properties:
  Duration: 0:00:00.000000000
  Seekable: yes
  Live: no
  video #0: H.264 (Constrained Baseline Profile)
    Stream ID:
    92ec0aada08d35b7679f87a90465b4cf955fbad11d2c853sec
    ad7e0568ac4a3a
    Width: 176
    Height: 144
    Depth: 24
    Frame rate: 0/1
    Pixel aspect ratio: 1/1
    Interlaced: false
    Bitrate: 0
    Max bitrate: 0
```

Index

- **Introduction**
 - Motivation
 - Overview
 - Building blocks
- **Features**
 - Modes: Launch, Create, Connect
 - UI panels
- **Architecture**
 - GRest
 - TypeScript bindings

Introduction

Motivation | Overview | Building blocks

Motivation

- Create a **didactic** tool for GStreamer newcomers
 - Visualize running programs
 - Create pipelines to validate their knowledge
- Create a **useful** tool for developers and QA
 - Once an application is deployed, it is difficult to debug, then debug remotely
- Make it UX/UI **attractive**, web ecosystem stands out

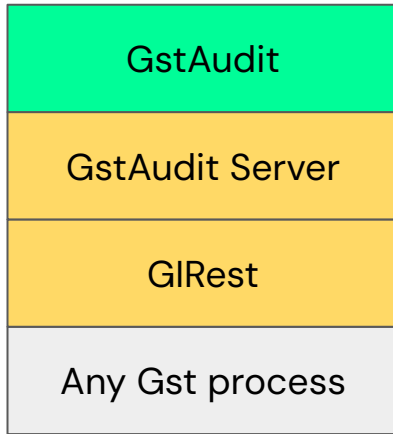
Overview

- Tool to audit, interact and visualize **running** GStreamer pipelines
- Similar tools exist: NVIDIA's Graph Composer, GstPipelineStudio, among others
- Built on top of **GIRest** (what?)
- Developed as part of the **Innovation Week @ Fluendo in 2025**
- Public at <https://github.com/fluendo/gst-audit/>
- Blog at <https://fluendo.com/blog/gst-audit-the-instrumentation-tool-for-your-pipelines/>



■ Building blocks

- Frontend **decoupled** from the backend
- REST communication
- **Frontend:**
 - TypeScript based
 - Next.js + TypeScript + React + React Flow (graph)
 - [Next.js](#) acts as a server too for callbacks
- **Backend**
 - Python based
 - Pygobject (introspection)
 - Apispec (python writer of OpenAPI)
 - Connexion (OpenAPI to python endpoints)
 - Frida to instrument the running process



 Frontend

 Backend

Features

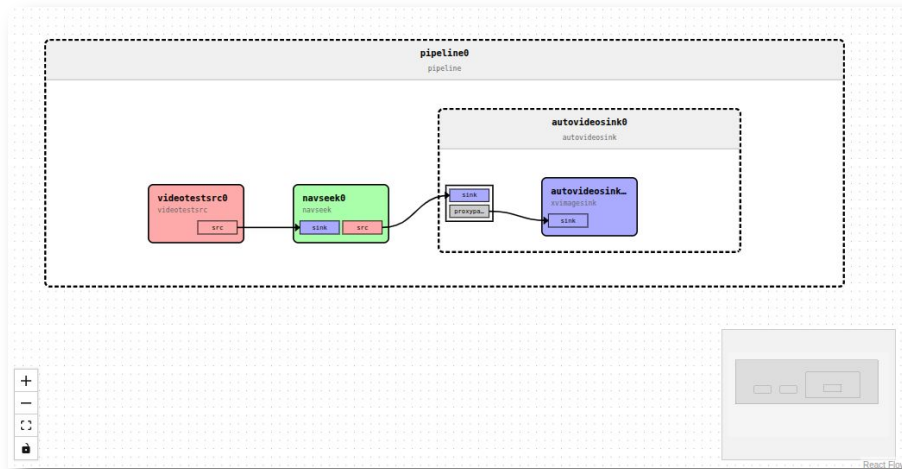
Modes | UI Panels

■ Modes: Launch, Create, Connect

- Launch
 - Parse the command line, instantiates a new gst-launch
 - Connects to the PID
 - `gstaudit_server.app launch "videotestsrc ! navseek ! autovideosink"`
- Connect
 - Connects to any running process
 - Detects a memory pattern (GObject type) to identify the GstPipelines or GstPipeline base classes (GstPlayBin3, etc)
 - `gstaudit_server.app connect <PID>`
- Create (WIP)
 - Empty binary placeholder linked with GStreamer libraries
 - Empty canvas to start creating your own pipeline
 - `gstaudit_server.app create`

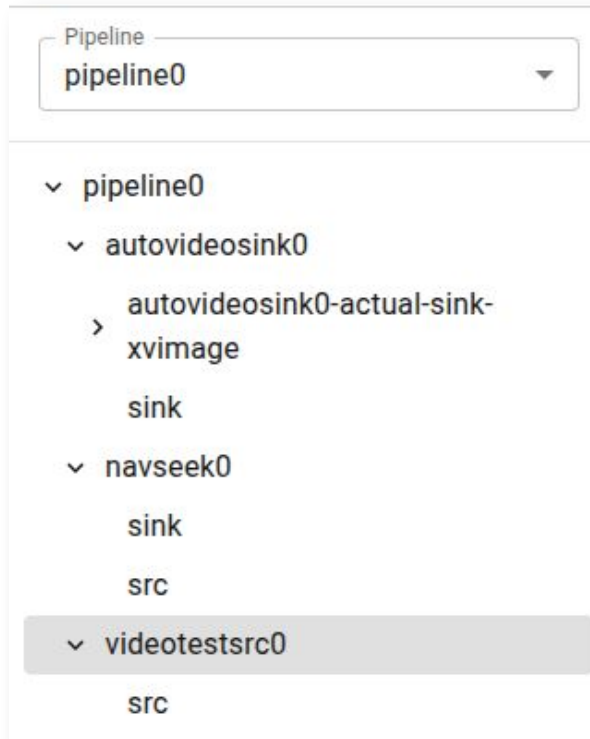
■ UI Panels | Pipeline graph

- Classic canvas to zoom in/out and pan
- Move elements
- **Not linking support yet, only view mode**
- **No parenting support yet**



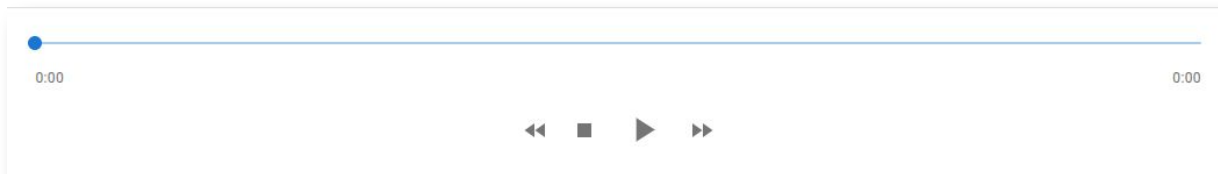
■ UI Panels | Pipeline and element selector

- See available pipelines in the process
- Navigate the tree of the pipeline: both elements and pads
- **Not linked pads are not shown**



■ UI Panels | Timeline and playback controls

- Classic playback controls: Play/Pause, Stop, Etc
- **No trickmodes support yet**



■ UI Panels | Element properties

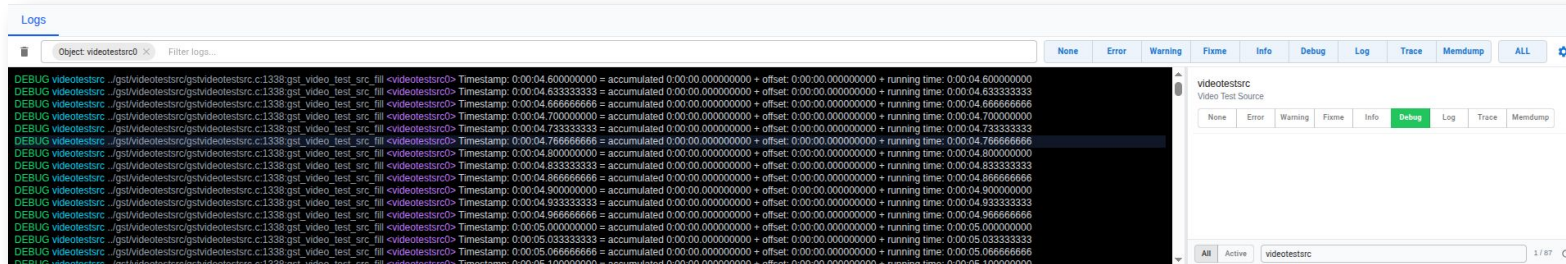
- Support Read-only properties
- Modify properties
- Supports main types: int, string, boolean, long, float, double ... and **samples!**

The image shows a screenshot of a software interface with a list of properties. Each property has a title, a description, and a control element. The properties are: 'parent' (read-only text), 'blocksize' (text input with value 4096), 'num-buffers' (text input with value -1), 'typefind' (radio button for False), 'do-timestamp' (radio button for False), and 'automatic-eos' (radio button for True).

Property Name	Description	Value / Control
parent	Parent The parent of the object <object at 0x63df7455bd10> Object properties are read-only	Read-only text
blocksize	Block size Size in bytes to read per buffer (-1 = default)	4096
num-buffers	num-buffers Number of buffers to output before sending EOS (-1 = unlimited)	-1
typefind	Typefind Run typefind before negotiating (deprecated, non-functional)	☐ False
do-timestamp	Do timestamp Apply current stream time to buffers	☐ False
automatic-eos	Automatic EOS Automatically EOS when the segment is done	☒ True

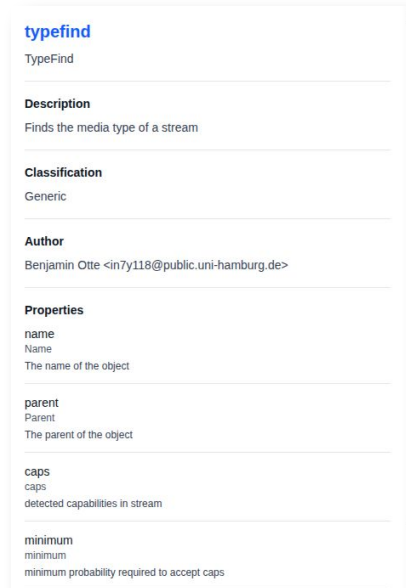
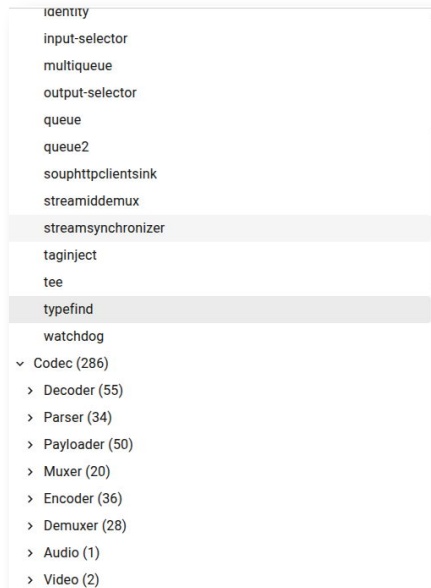
UI Panels | Logging

- Filter by categories
- Filter by elements
- **It is fast**



UI Panels | Factories

- List every factory available
- Inspect every factory for properties, pads, etc
- **Classification is not standard**
- **Part of the future editor**



Architecture

GIRest | Future

GIRest

- Bindings are generated automatically with PyGObject
- PyGObject reads the Typelib not the GIR file
 - `introspectable=0` are not represented in the Typelib
 - Typelib is platform specific not agnostic
 - Several class/methods/definitions do not follow a common pattern and need to be handled specifically (GstMiniObject?)
- Swagger available <http://localhost:9000/girest/ui> and <http://localhost:9000/gstaudit/ui>
- Nothing would have been possible without [FRIDA!](#)

Objects

- Objects are OpenAPI schemas under [components.schemas](#)
- Inheritance handled with “[AllOf](#)” directive
- Metadata (x-gi-) to identify namespace, class, type and name
- Base class has a pointer field

```
"GstElement": {  
  "type": "object",  
  "x-gi-type": "object",  
  "allOf": [{ "$ref": "#/components/schemas/GstObject" }]  
  "x-gi-type": "object",  
  "x-gi-namespace": "Gst",  
  "x-gi-name": "Element",  
  "x-gi-class": "GstElementClass"  
}
```

```
"GObjectTypeInstance": {  
  "x-gi-type": "struct",  
  "x-gi-namespace": "GObject",  
  "x-gi-name": "TypeInstance",  
  "type": "object",  
  "properties": {  
    "ptr": {  
      "$ref": "#/components/schemas/Pointer"  
    }  
  },  
  "required": [  
    "ptr"  
  ]  
}
```

■ Structs and recrods

- Objects are OpenAPI schemas under [components.schemas](#)
- Same with objects, all structs have a pointer field

```
"GstToc": {
  "x-gi-type": "struct",
  "x-gi-namespace": "Gst",
  "x-gi-name": "Toc",
  "type": "object",
  "properties": {
    "ptr": {
      "$ref": "#/components/schemas/Pointer"
    }
  },
  "required": [
    "ptr"
  ]
}
```

Enums and flags

- Enums as string enums with `x-gi-type: "enum"`
- Flags as string enums with `x-gi-type: "flags"`

```
{
  "GstState": {
    "type": "string",
    "enum": ["void-pending", "null", "ready", "paused", "playing"],
    "x-gi-type": "enum"
  }
}
```

Methods and functions

- Support for input, output and in/out parameters
- Each endpoint located at the corresponding class path
- Lifecycle management with `x-gi-transfer` metadata
- Nullable with `x-gi-null` metadata

```

"/Gst/version_string": {
  "get": {
    "summary": "",
    "description": "",
    "operationId":
"Gst--version_string",
    "tags": [
      "Gst"
    ],
    "parameters": [],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "application/json": {
            "schema": {
              "type": "object",
              "properties": {
                "return": {
                  "type": "string",
                  "x-gi-transfer":
"full",
                  "x-gi-null": false
                }
              }
            }
          }
        }
      }
    }
  }
}

```

```

"/Gst/version": {
  "get": {
    "parameters": [],
    "responses": {
      "200": {
        "description": "Success",
        "content": {
          "application/json": {
            "schema": {
              "type": "object",
              "properties": {
                "major": {
                  "type": "integer"
                },
                "minor": {
                  "type": "integer"
                },
                "micro": {
                  "type": "integer"
                },
                "nano": {
                  "type": "integer"
                }
              }
            }
          }
        }
      }
    }
  }
}

```

Callbacks I

- Callbacks follow the OpenAPI definition for [callbacks](#)
- Each callback parameter is a query parameter
- Each callback has a schema for its body and return value in `components.schema`
- Scope types are managed by the `x-gi-callback-style`
 - Sync: Response 200 with the return schema if any. If response needed, the server waits on it.
 - Async: Response 204. Notification only, no actual return value.

```
{
  "name": "func",
  "in": "query",
  "required": true,
  "schema": { "type": "string", "format": "uri",
    "description": "URL to invoke for StructureForeachFunc
callback" },
  "x-gi-callback": "#/components/schemas/GstStructureForeachFunc",
  "x-gi-callback-style": "sync"      // or "async"
}
```

```
"callbacks": {
  "StructureForeachFunc": {
    "{$request.query.func}": {
      "post": {
        "requestBody": { "content": { "application/json": {
          "schema": { "$ref":
            "#/components/schemas/GstStructureForeachFunc" } } } },
        "responses": { "200": { "content": { "application/json": {
          "schema": { "$ref":
            "#/components/schemas/GstStructureForeachFuncReturn" } } } } },
        "security": [ { "callbackSignature": [] } ]
      }
    }
  }
}
```

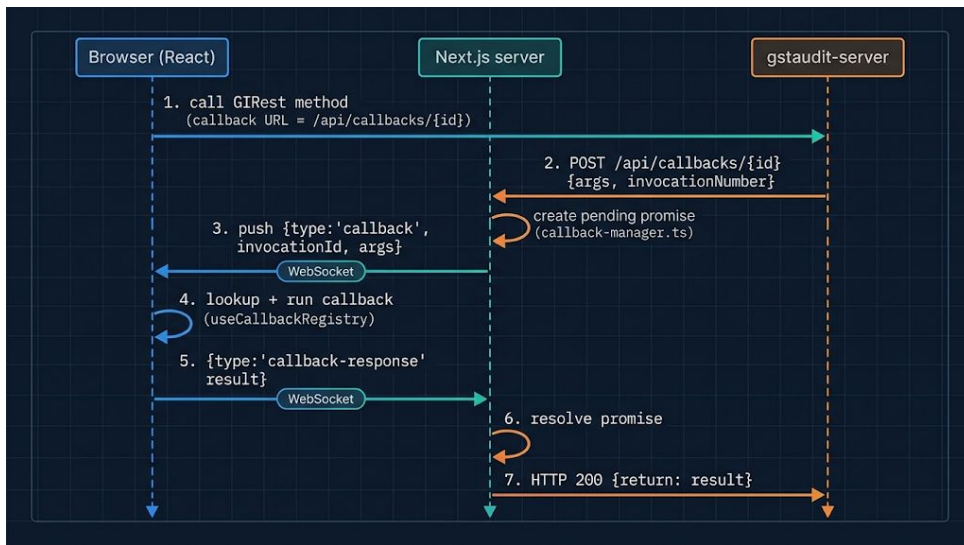
Callbacks II

- Each callback is stored with the corresponding session id (client) and a shared secret (HMAC)
- Each callback parameter is defined as a property in the schema

```
{
  "type": "object",
  "x-gi-type": "callback",
  "x-gi-namespace": "Gst",
  "properties": {
    "field_id": { "type": "integer" },
    "value": { "$ref": ".../GObjectValue" },
    "user_data": { "$ref": ".../Pointer" },
    "sessionId": { "type": "string" },
    "callbackName": { "type": "string" },
    "args": { "type": "array" },
    "invocationNumber": { "type": "integer" },
    "timestamp": { "type": "string", "format": "date-time" }
  },
  "required": ["sessionId", "callbackName", "args"]
}
```

Callbacks III

- Multi-client based on a session id
- [Next.js](#) routes the call the WebSocket and blocks if it needs a response
- The browser via WebSocket At the client side, callbacks are handled via WebSockets in the TypeScript/JS layer

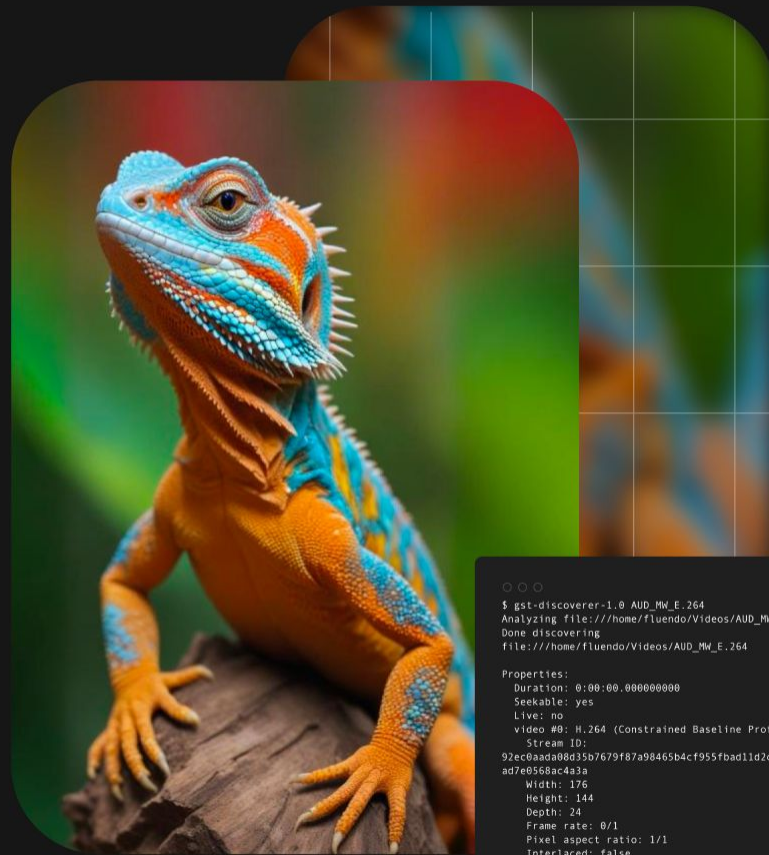


Future

- Normalize several API inconsistencies
- Logs don't have a serializer and lack of a callback removal
- Extend property types to enhance the UI: Rectangle, Color, etc
- Normalize the factories categorization
- Implement a real editor to connect and parent/unparent elements
- Proxy the sink content
- Intercept the traveling buffers
- **Tell me your feature!**



Stop taking risks.
Start finding
solutions.
Thank you!



```
$ gst-discoverer-1.0 AUD_MW_E.264
Analyzing file:///home/fluendo/Videos/AUD_MW_E.264
Done discovering
file:///home/fluendo/Videos/AUD_MW_E.264

Properties:
  Duration: 0:00:00.000000000
  Seekable: yes
  Live: no
  video #0: H.264 (Constrained Baseline Profile)
    Stream ID:
    92ec0aada08d35b7679f87a98465b4cf955fbad11d2c8535ec
    ad7e0568ac4a3a
    Width: 176
    Height: 144
    Depth: 24
    Frame rate: 0/1
    Pixel aspect ratio: 1/1
    Interlaced: false
    Bitrate: 0
    Max bitrate: 0
```