bogJS – A JavaScript framework for object-based rendering in browsers

Michael Weitnauer & Michael Meier

ORPHEUS Project

- EU-funded Horizon 2020 project
- December 2015 to June 2018 (2.5 years)
- Will develop, implement and validate an **end-to-end object-based media chain**
- 10 partners

Fraunhofer IIS

b com

BR

BBC R&D

ircam Centre Pompidou

TRINNOV AUDIO

[°°] Elephantcandy

IRT

MAGIX

EURESCOM

ORPHEUS will…

ORPHEUS
Object-Based Audio Experience

- □ … develop a concept for the **transition of existing infrastructure**, systems and software/tools to a regular operational service of object-based audio

- □ … demonstrate a new, **prodigious user experience** through the creation of a workflow application for the use of object-based audio as an emerging future broadcast technology

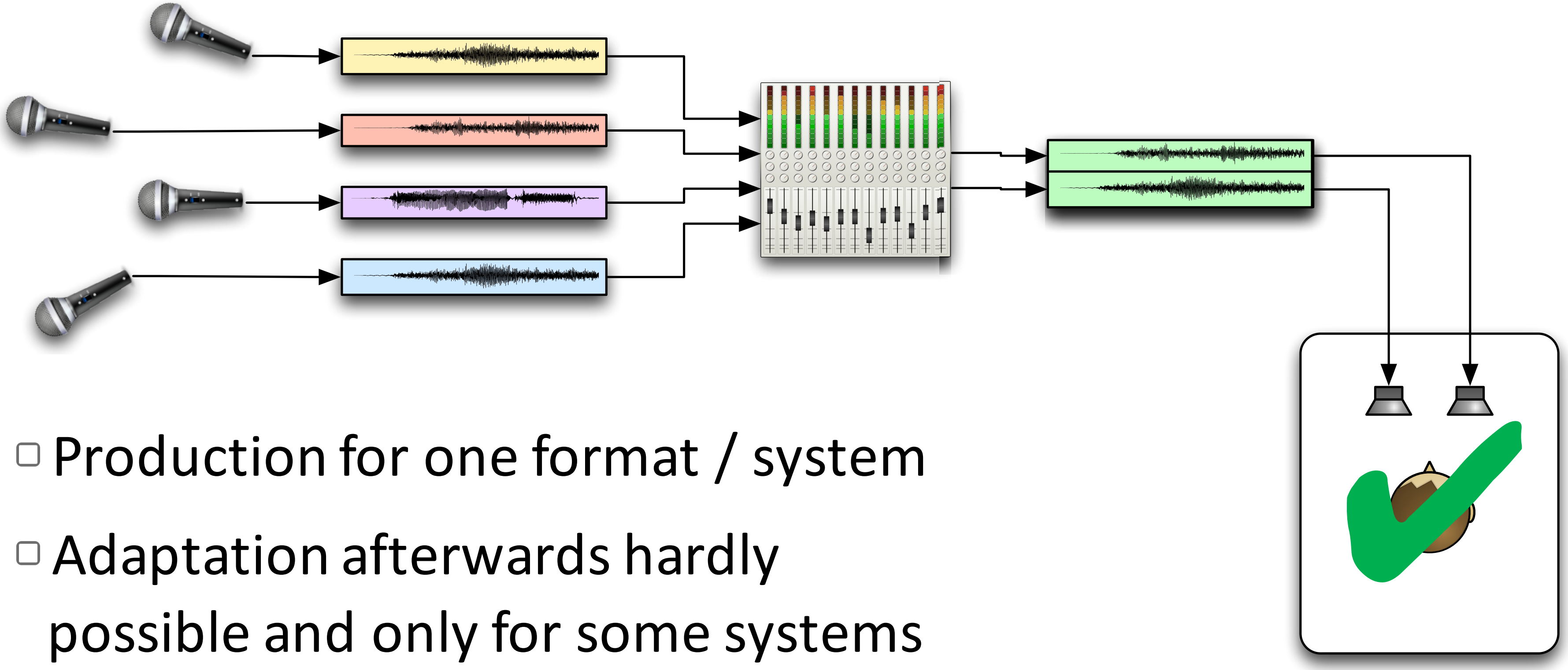
- □ … create a **reference architecture** and guidelines on how to implement an end-to-end broadcasting chain for object-based audio

What is ‘object-based audio’ ??

‘object-based’ != immersive audio

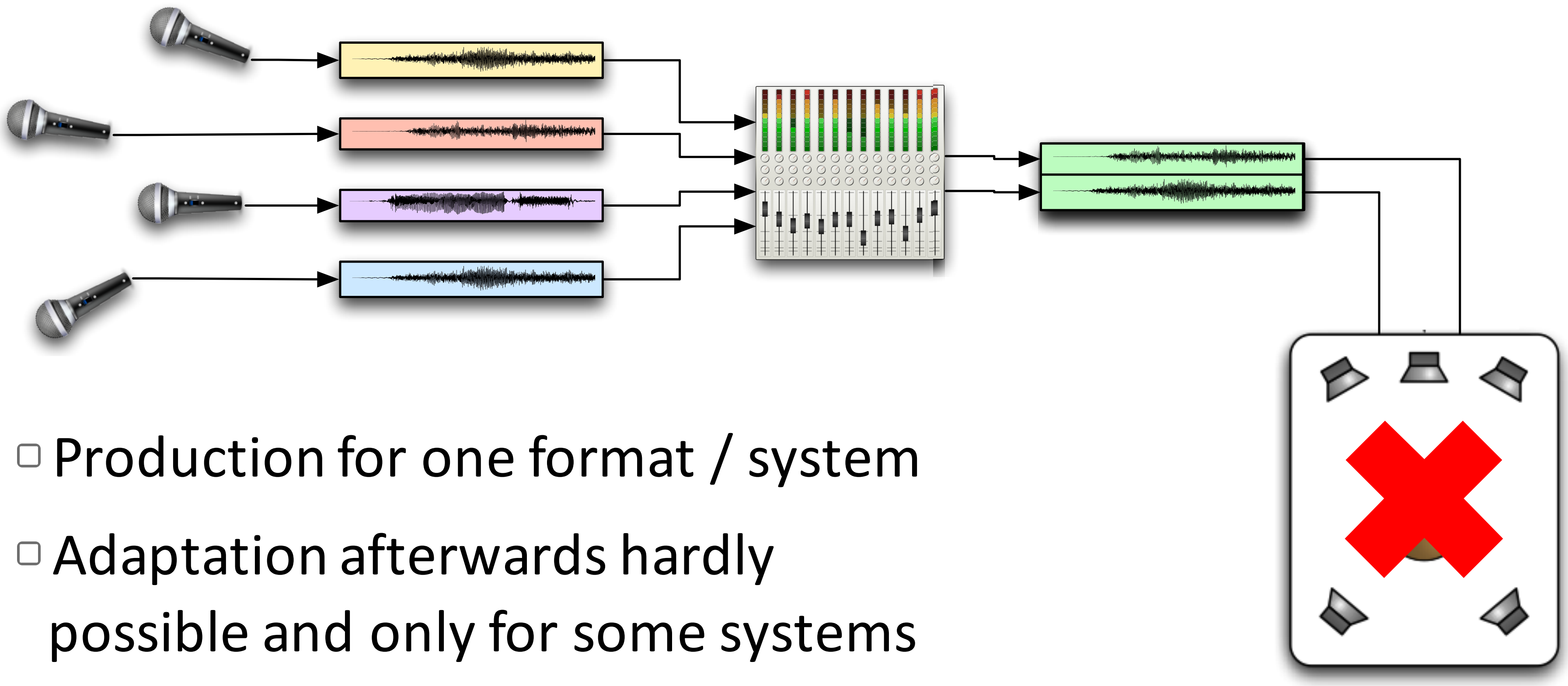
‘object-based’ == audio + metadata

Audio production and distribution workflow today (channel-based)



- Production for one format / system
- Adaptation afterwards hardly possible and only for some systems

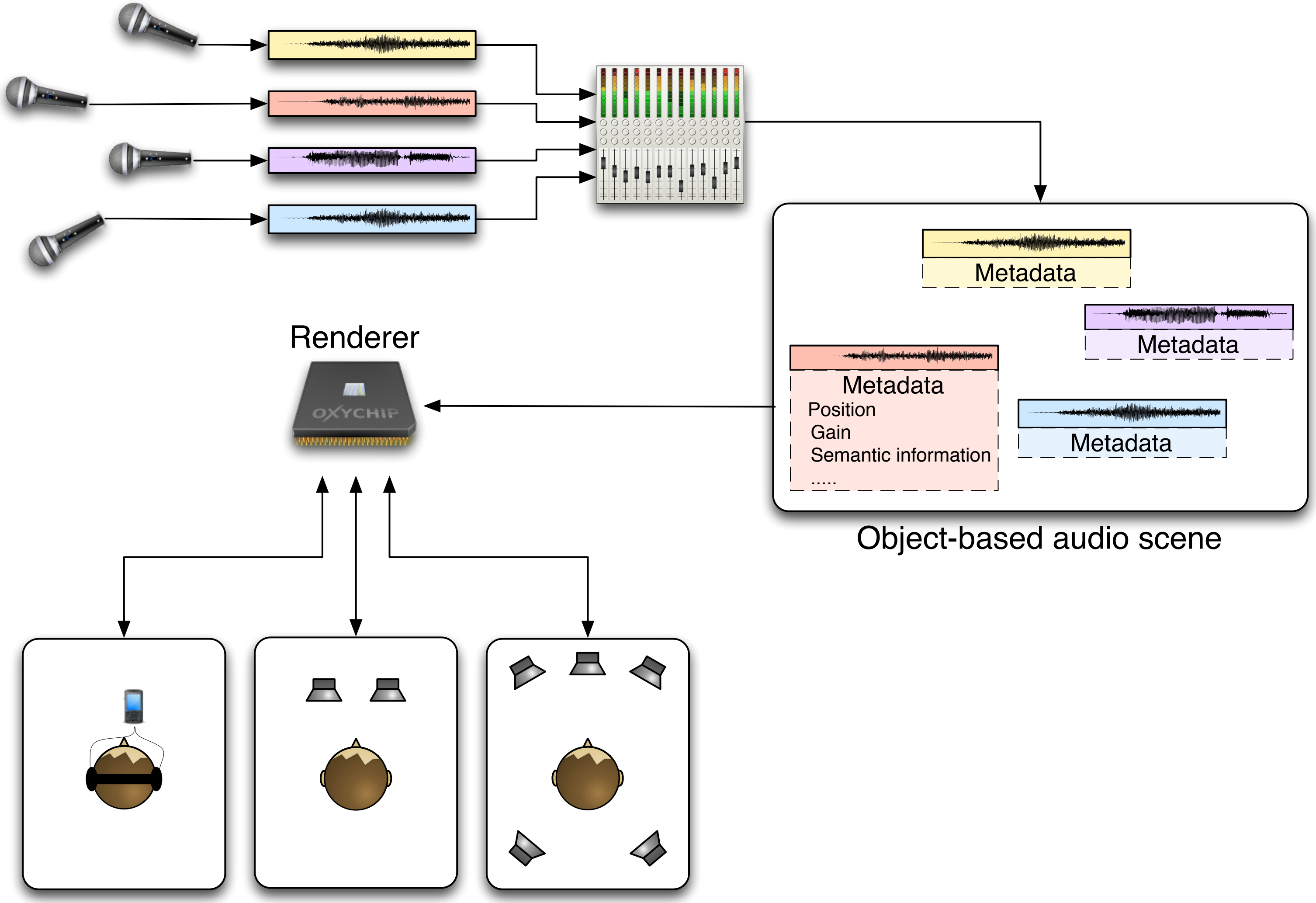
Audio production and distribution workflow today (channel-based)



- □ Production for one format / system
- □ Adaptation afterwards hardly possible and only for some systems

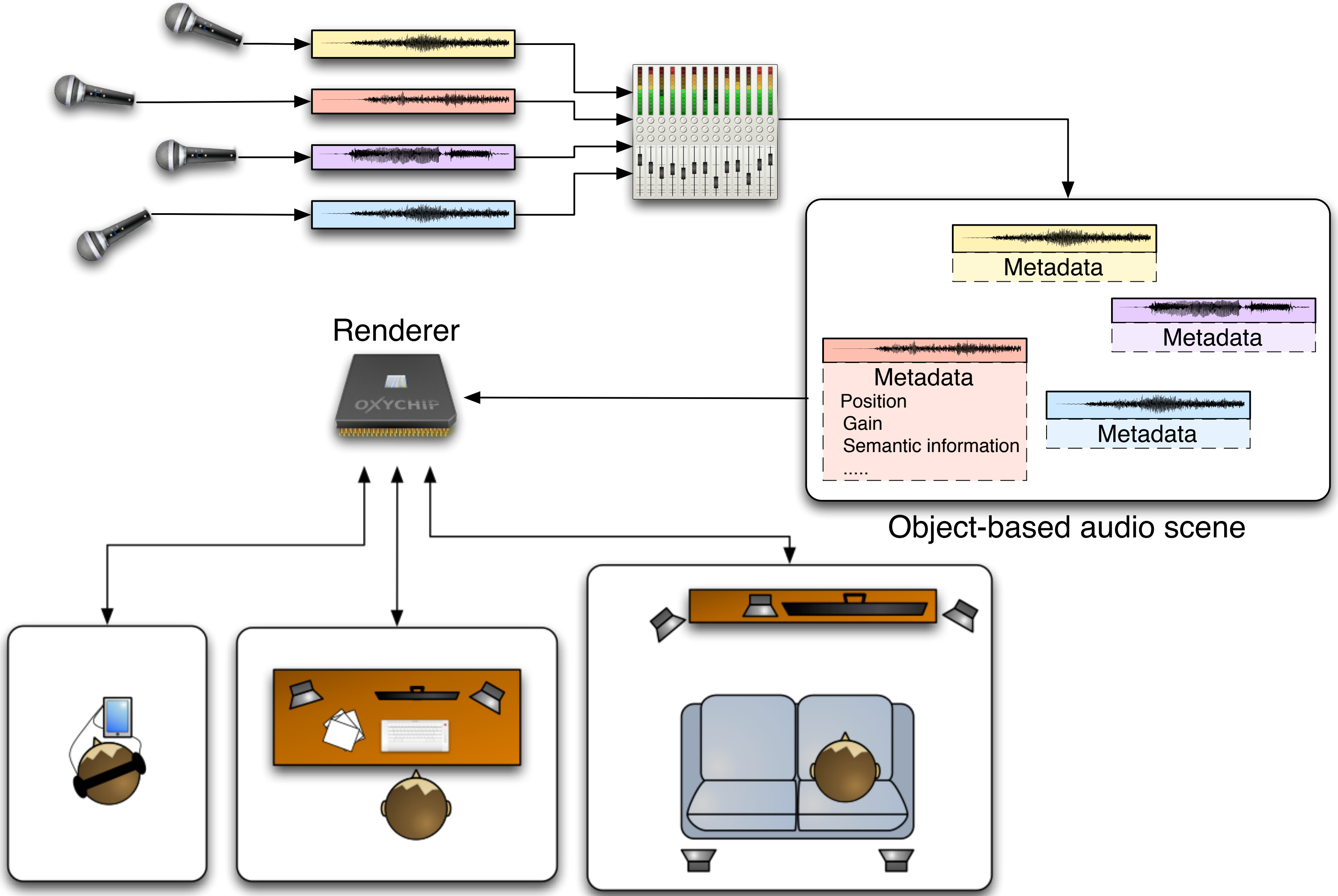
Object-based workflow

- □ Format agnostic
  - □ Adaptation to playback system
  - □ Support for future formats
- □ Accessibility
  - □ Multiple languages
  - □ Voice over
- □ Personalization
  - □ Personal mix (e.g. dialogue)
  - □ Variable length
- □ Interactivity



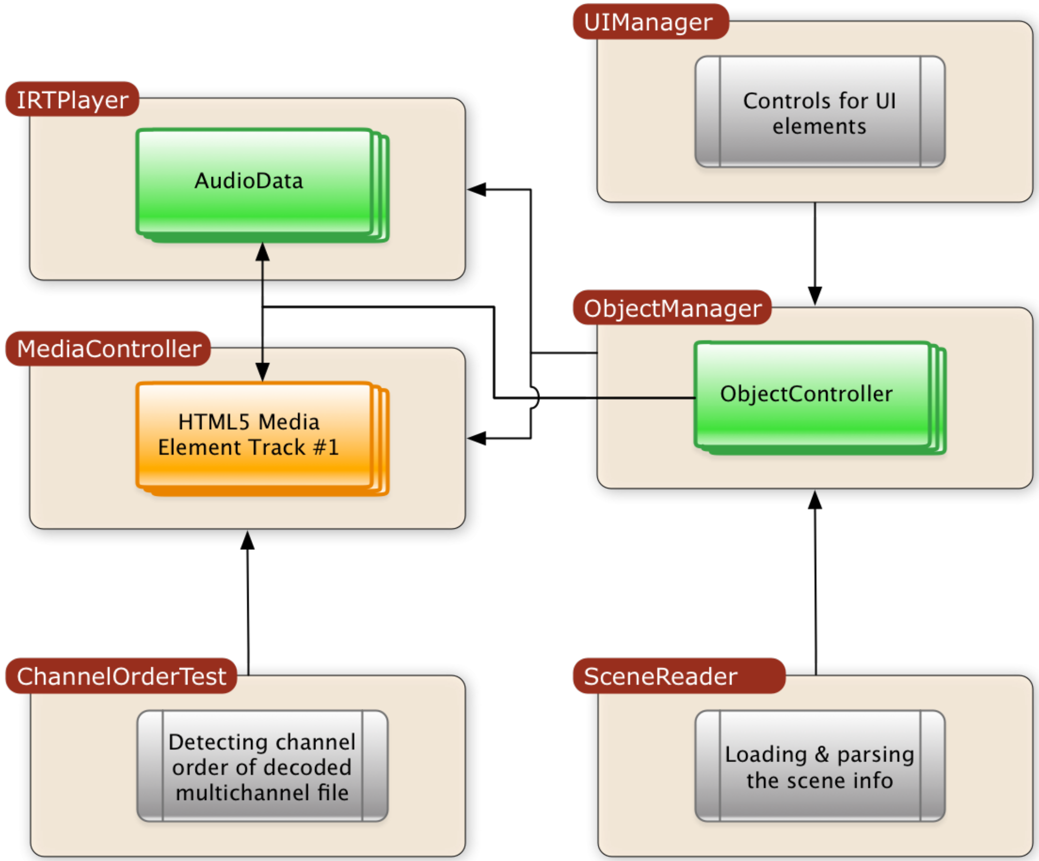
Object-based workflow

- □ Format agnostic
  - □ Adaptation to playback system
  - □ Support for future formats
- □ Accessibility
  - □ Multiple languages
  - □ Voice over
- □ Personalization
  - □ Personal mix (e.g. dialogue)
  - □ Variable length
- □ Interactivity



bogJS - Introduction

- JavaScript framework
- Published under MIT license
- Written in ES5
- Using many parts of the Web Audio API
- 3rd party libs used:
  - jQuery
  - jQuery mousewheel
  - jQuery UI
  - jQuery Transit
  - loglevel
  - underscore.js
  - WAAClock.js



bogJS – Basic Concepts & Usage

- All relevant info is read from a scene file
- Three options to load and play audio signals
  - Single audio object (< 3min, file)

    → AudioBufferSourceNode (XHR)
  - Multiple single objects grouped (< 3min, files)

    → AudioBufferSourceNodes
  - One or more longer objects with same duration (any duration, file / stream)

    → MediaElementSourceNode (<audio> / <video>)
- Currently implemented object descriptors: *Gain, Position, Interactive, Active*

```
<script src="dist/bogJS-latest.js"></script>
<!-- Only if you want to use the UI relevant parts: -->
<script src="dist/bogJS-ui-latest.js"></script>

<script>
    var om;
    $(document).on("om_ready", function(){
        om.start();
    });
    om = new ObjectManager("path/to/sceneFile.spatdif");
</script>
```

bogJS – Sample Scene (SpatDIF like)

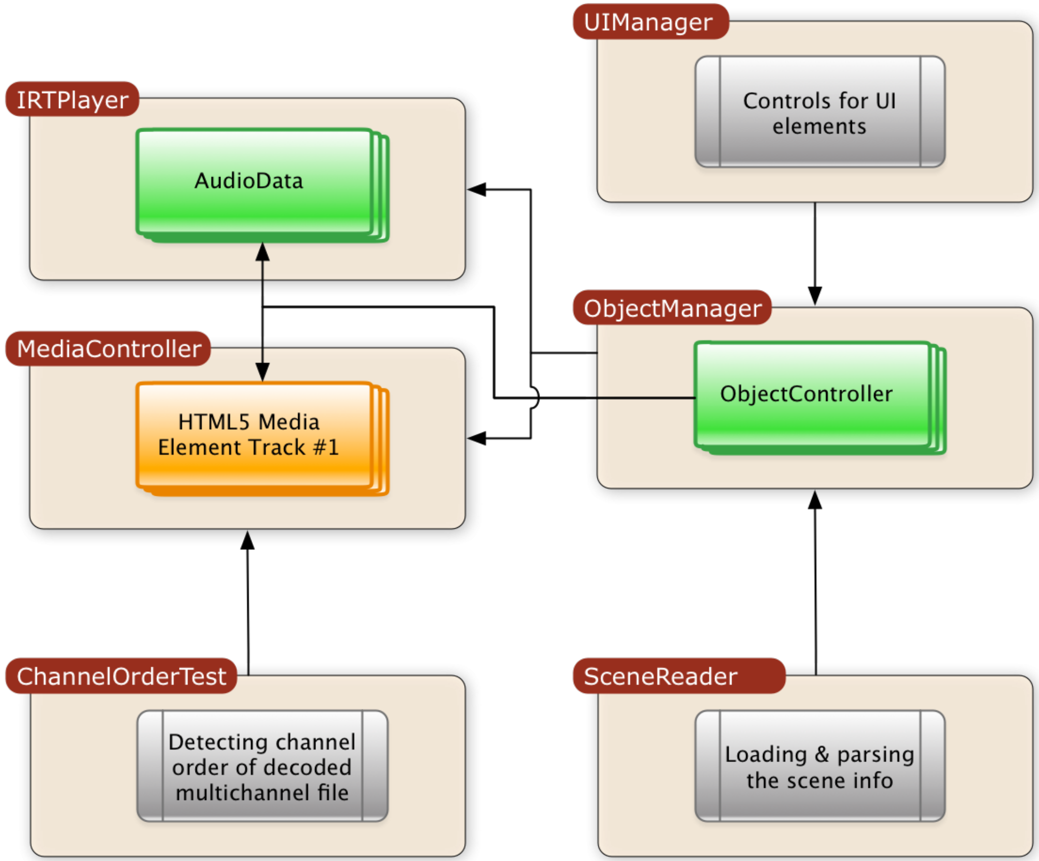
```
/spatdif/meta/audiobed/url  local/path/to/audiofile
/spatdif/meta/audiobed/tracks  5

/spatdif/time  0.0
/spatdif/source/Bed0/position  -1.0 0.0 -1.0
/spatdif/source/Bed0/active  1
/spatdif/source/Bed0/track_mapping  bed_0
/spatdif/source/Bed0/gain  1
/spatdif/source/Bed1/position  1.0 0.0 -1.0
/spatdif/source/Bed1/active  1
/spatdif/source/Bed1/track_mapping  bed_1
/spatdif/source/Bed1/gain  1
/spatdif/source/Bed2/position  -1.0 0.0 1.0
/spatdif/source/Bed2/active  1
/spatdif/source/Bed2/track_mapping  bed_2
/spatdif/source/Bed2/gain  1
/spatdif/source/Bed3/position  1.0 0.0 1.0
/spatdif/source/Bed3/active  1
/spatdif/source/Bed3/track_mapping  bed_3
/spatdif/source/Bed3/gain  1
/spatdif/source/Speech/position  0.05 0.0 -2.52
/spatdif/source/Speech/active  1
/spatdif/source/Speech/track_mapping  bed_4
/spatdif/source/Speech/gain  1

/spatdif/time  70.754
/spatdif/source/Birds1_L/position  -0.86 0.0 -1.77
```

```
/spatdif/source/Birds1_L/interpolate  0
/spatdif/source/Birds1_L/active  1
/spatdif/source/Birds1_L/track_mapping  http://irt.de/birds1_l
/spatdif/source/Birds1_L/gain  1
/spatdif/source/Birds1_L/interactive  0
/spatdif/source/Birds1_R/position  0.86 0.0 -1.79
/spatdif/source/Birds1_R/interpolate  0
/spatdif/source/Birds1_R/active  1
/spatdif/source/Birds1_R/track_mapping  http://irt.de/birds1_r
/spatdif/source/Birds1_R/gain  1
/spatdif/source/Birds1_R/interactive  0
/spatdif/source/Stones_R/gain  0.75

/spatdif/time  120.0
/spatdif/source/Stones_L/position  -0.603  0.0  -0.685
/spatdif/source/Stones_L/active  0
/spatdif/source/Stones_R/position  0.291  0.0  -0.771
/spatdif/source/Stones_R/gain  0.5
/spatdif/source/Stones_L/group  Stones

/spatdif/time  123.8
/spatdif/source/Stones_L/position  -0.5845  0.0  -0.6325
/spatdif/source/Stones_L/active  1
/spatdif/source/Stones_R/position  0.2915  0.0  -0.7165
/spatdif/source/Stones_R/gain  0.25
/spatdif/source/Stones_R/group  Stones
```

bogJS – Further Basic Concepts

- ▢ Time changes are realized with WAAClock
- ▢ The UIManager class provides basic functionality for a 2D user interface
- ▢ For longer files or streams (e.g. to be used as audio-beds) MediaController class should be used

→ Problem: decoded channel order diversity of browsers



bogJS – ChannelOrderTest

- Can be used to detect automatically the order of the decoded multichannel tracks
- Frequency detection with AnalyserNode
- Provided test files
    - WAV files from 2 - 12 channels with a duration of 1 second
    - For each channel, an increased sinus tone frequency is used to detect the order after the encoding and decoding

```
/* Simple example */
var ch = new ChannelOrderTest("mp4", 6);
ch.testChs();
// -> [0, 2, 1, 4, 5, 3]
```

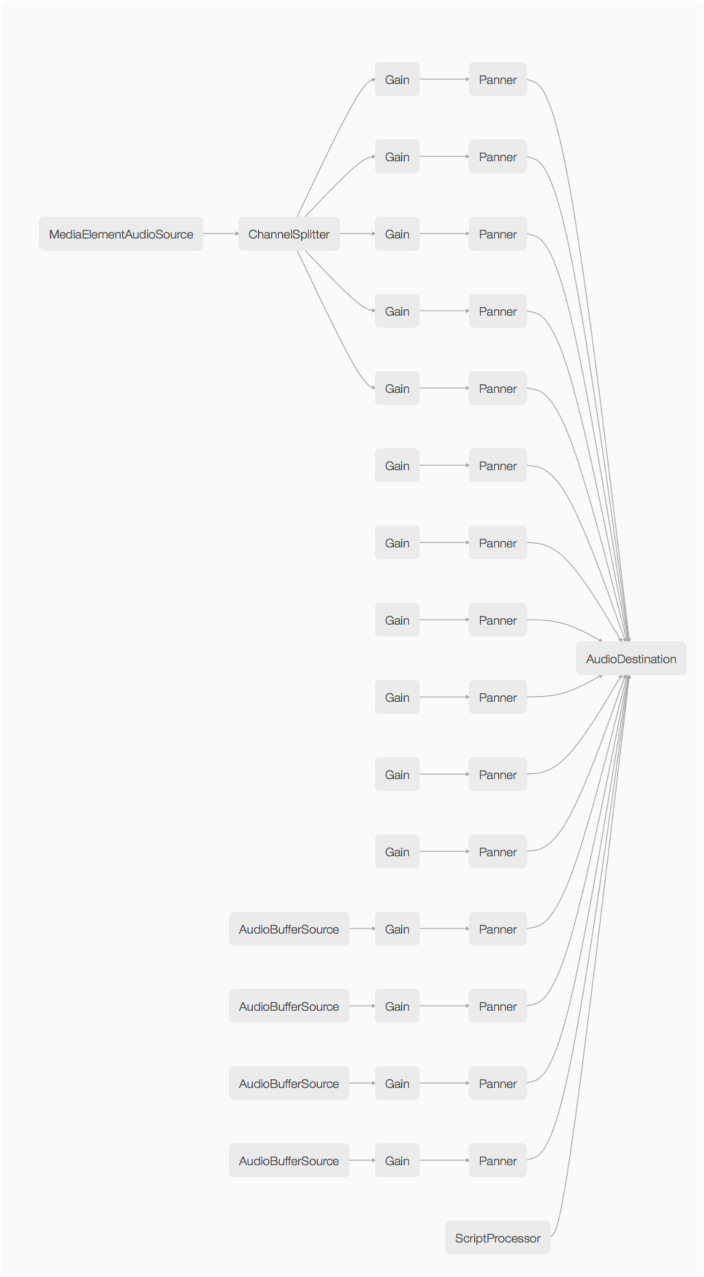
```
/* Advanced example */
var ctx = new AudioContext();
var ch;
$(ch).on('order_ready', function(e, order){
    console.log("Got channel order: " + order);
    doSomething();
});
ch = new ChannelOrderTest("ogg", 4, ctx,
"path/to/testfiles/");
// -> Got channel order: [0, 3, 2, 1]
```

bogJS - Demo

© Tony Hall

bogJS - Summary

- □ ... has multiple options to load and play audio signals
- □ ... offers automatic detection of decoded multichannel track order
- □ ... can render scenes for Stereo (2.0) and binaural (headphones) output
- □ ... works on all modern browsers which support the Web Audio API



bogJS – Plans for the future

- Implementation of current and upcoming standards, especially ITU-R BS.2076 (ADM)
- JSON representation of ADM data
- Integration of upcoming SpatialPannerNode
- Providing more interpolation options by making use of the AudioParam interface
- Extend streaming capabilities

Thank you!

Michael Weitnauer
@ weitnauer@irt.de
@_mweitnauer

Michael Meier
@ meier@irt.de
@mikedevnull

www.irt.de/en
@IRTpresse

www.orpheus-audio.eu
@ORPHEUS_Audio

https://github.com/IRT-Open-Source/bogJS