



Professur  
Allgemeine Nachrichtentechnik

*Prof. Dr.-Ing. Udo Zölzer*



HELMUT SCHMIDT  
UNIVERSITÄT

# BeagleJS: HTML5 and JavaScript based Framework for the Subjective Evaluation of Audio Quality

**Sebastian Kraft, Udo Zölzer**

**Helmut-Schmidt-University, Hamburg**

May 2nd, 2014

Objective evaluation of audio quality is important in many fields:

- audio codecs
- effects
- hardware
- ...

But...

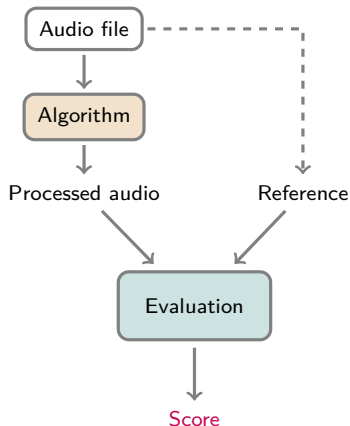
- what is good audio quality?
- is there any objective evaluation method?
- how to measure audio quality?

- Audio quality evaluation
- Listening test methodology
- The BeagleJS framework
- Demo
- Advanced usage

- Audio quality evaluation
- Listening test methodology
- The BeagleJS framework
- Demo
- Advanced usage

# Audio quality evaluation

## General approach



- processed audio and unprocessed reference as input
- resulting score should be proportional to human perception
- mapping between numerical score and perceived quality

# Audio quality evaluation

## Simple physical measures

---



HELMUT SCHMIDT  
UNIVERSITÄT

- signal to noise ratio
- time and frequency response
- total harmonic distortion
- mean square error to reference signal



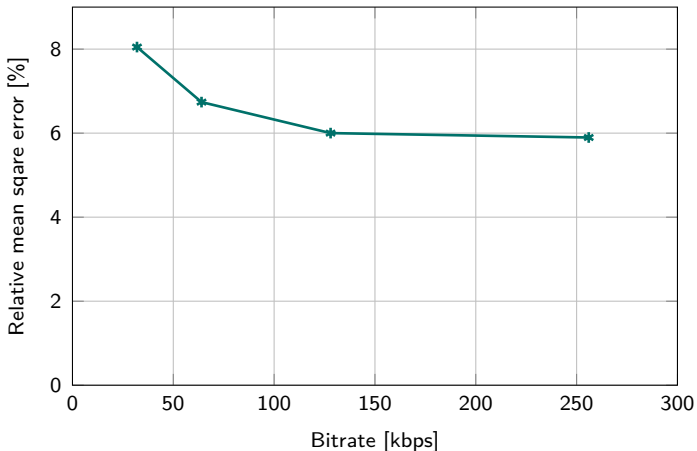
# Audio quality evaluation

## Simple physical measures



HELMUT SCHMIDT  
UNIVERSITÄT

Relative mean square error over Bitrate for MP3



# Audio quality evaluation

## Simple physical measures

---



- signal to noise ratio
  - time and frequency response
  - total harmonic distortion
  - mean square error to reference signal
- 
- ⊕ simple and efficient calculation
  - ⊕ well suited for automatic evaluations with huge test sets
  - ⊖ bad correlation with human perception



# Audio quality evaluation

## Perceptually motivated measures

---

- new compound measures that try to mimic human hearing
- consider psycho-acoustic and biological aspects
  - masking effect
  - frequency-dependent hearing threshold
  - noise-to-mask ratio
- established tools for various applications PEAQ (audio codecs), PESQ (speech), PEASS (source separation)

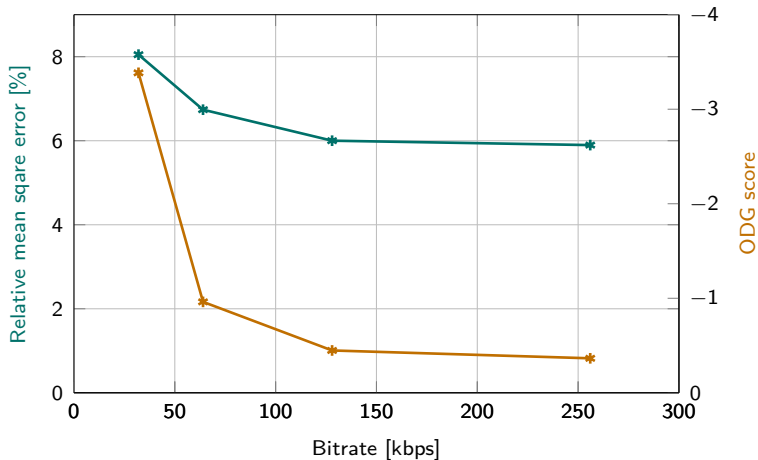
# Audio quality evaluation

## Perceptually motivated measures



HELMUT SCHMIDT  
UNIVERSITÄT

PEAQ ODG score over Bitrate for MP3



# Audio quality evaluation

## Perceptually motivated measures

---

- new compound measures that try to mimic human hearing
  - consider psycho-acoustic and biological aspects
    - masking effect
    - frequency-dependent hearing threshold
    - noise-to-mask ratio
  - established tools for various applications PEAQ (audio codecs), PESQ (speech), PEASS (source separation)
- 
- ⊕ usually good correlation with human perception
  - ⊕ well suited for automatic evaluations with huge test sets
  - ⊖ need for additional subjective listening tests for verification
  - ⊖ complex implementation

# Audio quality evaluation

## Subjective evaluation methods

---

- "listening tests"
  - present processed signals and unprocessed references to a listener
  - listener should rate the similarity or audio quality of the signals
- 
- ⊕ perfect correlation with human perception
  - ⊖ very time-consuming
  - ⊖ only for small test sets
  - ⊖ not reproducible

- Audio quality evaluation
- **Listening test methodology**
- The BeagleJS framework
- Demo
- Advanced usage

Humans are rarely objective in their judgements

- it is difficult to retrieve **objective** results from a **subjective** evaluation
  
- use standardised test methods
- ITU provides several helpful publications
  - ITU-R Recommendation BS.1284-1, general methodology
  - ITU-R Recommendation BS.1116-1, ABC tests
  - ITU-R Recommendation BS.1534-1, MUSHRA tests
  - ...
  
- a sufficiently big number of participants will reduce the risk of statistical outliers and errors

# Listening test methodology

## Layout



HELMUT SCHMIDT  
UNIVERSITÄT

### Mushra Demo Test

#### Castanets (2 of 2)

Reference	Play	Stop	Bad	Poor	Fair	Good	Excellent
Test Item 1	Play	Stop					
Test Item 2	Play	Stop					
Test Item 3	Play	Stop					
Test Item 4	Play	Stop					
Test Item 5	Play	Stop					

Previous Test

Next Test

00:03



Loop

Volume

- items should appear in random order
- neutral names
- controls for start/pause of playback
- possibility to loop short parts

- use items that underline various aspects of the tested algorithms
- items should be short ( $< 10$  s)
- overall time for test should not exceed 15 min
- SQAM CD for example provides a wide variety of test signals (speech, instruments, synthetic signals, ...)





- background and experience of participants can influence the results
- recommended number of participants by the ITU:
  - at least 10 expert listeners
  - more than 20 non-expert listeners
- use of anchor signals and hidden references to check validity
- well-defined listening conditions (headphones, ...)

# Listening test methodology

## Summary

---



- test design is crucial
- many pitfalls on the way to significant results
- ...
- difficult to reach an adequate number of participants
- finally, how to **create** a listening test?!?

- test design is crucial
- many pitfalls on the way to significant results
- ...
- difficult to reach an adequate number of participants
- finally, how to **create** a listening test?!?

⇒ BeaqleJS

- Audio quality evaluation
- Listening test methodology
- **The BeaqleJS framework**
- Demo
- Advanced usage

- based on HTML5 and JavaScript
- no dependencies to additional plugins like Adobe Flash
- use any modern web browser as a runtime environment
- two pre-defined test schemes: ABX and MUSHRA
- flexible usage scenarios
  - simple world-wide distribution of large-scale tests
  - distribute it locally on your workgroup server in the intranet
  - supervised tests on single computers, tablets, ...
  - perform self-blind tests

```
<!-- basic usage -->
<audio src="audio.mp3" />

<!-- advanced usage -->
<audio controls autoplay preload="auto">
  <source src="audio.mp3" type='audio/mpeg'>
  <source src="audio.ogg" type='audio/ogg'>
  <p>HTML5 audio is not supported by your browser!</p>
</audio>
```

- complete JavaScript API for `play()`, `pause()`, `load()` and various callbacks
- browser can show basic player controls
- create own controls with HTML and Javascript

	Internet Explorer	Firefox	Chrome	Opera	Safari
WAV PCM	no	> 3.5	yes	> 11.00	> 3.1
Ogg Vorbis	no	> 3.5	yes	> 10.50	with XiphQT
MP3	> 9.0	> 26, not OS X	yes	> 14	> 3.1
AAC	> 9.0	> 26, not OS X	yes	> 14	> 3.1

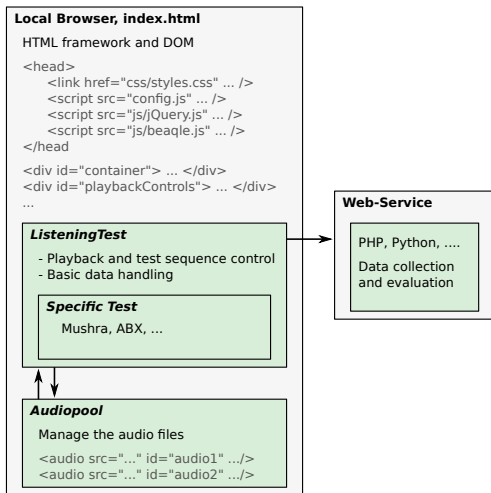
- codec support in the HTML5 <audio> element is browser dependant
- no lossless codec (e.g. FLAC) support with any browser
- best choice is WAV PCM with 16 bit, 44.1 kHz

# BeaqlJS

## Block diagram



HELMUT SCHMIDT  
UNIVERSITÄT





# BeaqlJS

## Block diagram



HELMUT SCHMIDT  
UNIVERSITÄT

### Local Browser, index.html

HTML framework and DOM

```
<head>
  <link href="css/styles.css" ... />
  <script src="config.js" ... />
  <script src="js/jquery.js" ... />
  <script src="js/beaql.js" ... />
</head>
<div id="container"> ... </div>
<div id="playbackControls"> ... </div>
...
```

#### ListeningTest

- Playback and test sequence control
- Basic data handling

#### Specific Test

Mushra, ABX, ...

#### Audiopool

Manage the audio files

```
<audio src="..." id="audio1" .../>
<audio src="..." id="audio2" .../>
```

### Web-Service

PHP, Python, ....  
Data collection  
and evaluation

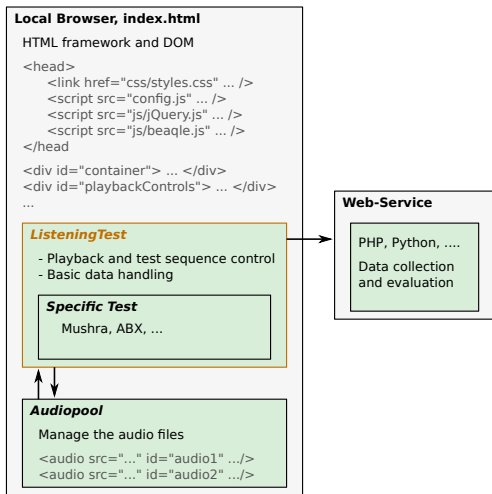
- standard HTML page
- no server-side dependencies
- includes all necessary files in the header
- defines a set of container `<div>`-tags
- contents and visibility are controlled by the `beaql.js` script

# BeaqlJS

## Block diagram



HELMUT SCHMIDT  
UNIVERSITÄT



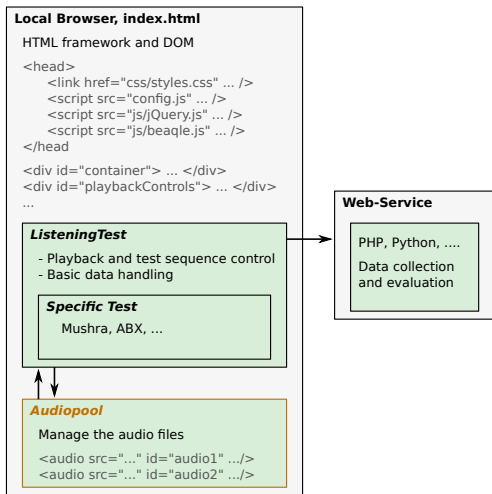
- ListeningTest base class
- general functionality
  - test sequence control
  - playback controls
  - data loading

# BeaqlJS

## Block diagram



HELMUT SCHMIDT  
UNIVERSITÄT



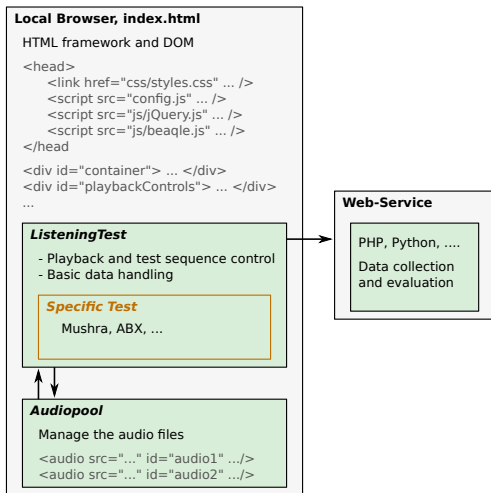
- Audiopool class
- manages loading of test items into <audio>-tags
- synchronized playback
- callback to main class when audio is ready

# BeaqlJS

## Block diagram



HELMUT SCHMIDT  
UNIVERSITÄT



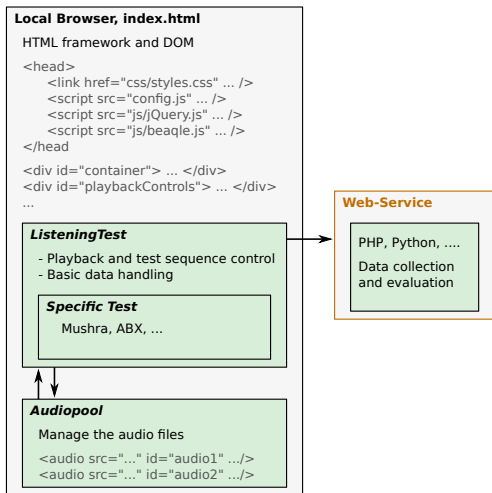
- test specific functionality
- layout and arrangement of test items
- evaluation and storage of results

# BeaqlJS

## Block diagram



HELMUT SCHMIDT  
UNIVERSITÄT



- interface to server-side scripts
- transmit results for centralised evaluation

- Audio quality evaluation
- Listening test methodology
- The BeaqleJS framework
- **Demo**
- Advanced usage

# ABX Listening Test



HELMUT SCHMIDT  
UNIVERSITÄT

## ABX Demo Test

Schubert (1 of 2)

A  X  B

Press buttons to start/stop playback.



Please select the item which is closest to X!

Previous Test

Next Test

00:04



Loop

Volume



# MUSHRA Listening Test

## ITU-R Recommendation BS.1534-1



HELMUT SCHMIDT  
UNIVERSITÄT

### Mushra Demo Test

#### Castanets (2 of 2)

Reference	Play	Stop	Bad	Poor	Fair	Good	Excellent
Test Item 1	Play	Stop	<input type="range"/>				
Test Item 2	Play	Stop	<input type="range"/>				
Test Item 3	Play	Stop	<input type="range"/>				
Test Item 4	Play	Stop	<input type="range"/>				
Test Item 5	Play	Stop	<input type="range"/>				

Previous Test

Next Test

00:03



Loop

Volume



- Audio quality evaluation
- Listening test methodology
- The BeagleJS framework
- Demo
- **Advanced usage**

- pack results into a JSON structure
- transmit them to a web service (AJAX HTTP request)
- store and analyse the results in a central place

### Future work:

- web interface to create and manage listening tests
- tools to evaluate and visualise results
- direct export to e.g. MATLAB, R, Python, ...

```
function MyTest(TestData) {
  ListeningTest.apply(this, arguments);
}
MyTest.prototype = new ListeningTest();
MyTest.prototype.constructor = MyTest;

MyTest.prototype.createTestDOM = ...
MyTest.prototype.saveRatings   = ...
MyTest.prototype.readRatings   = ...
MyTest.prototype.formatResults = ...
```

- create a new class MyTest and inherit from ListeningTest
- implement test specific functionality
  - createTestDOM()
  - saveRatings()
  - readRatings()
  - formatResults()

# Summary

## BeaqlJS

---

- run listening tests in any web browser
- flexible distribution to the participants
- easy to setup
- two pre-defined test schemes (ABX, MUSHRA)
- simple expansion
- collect and evaluate results using a central web service
- it is available for free

# Where can I get it?

---



[https://github.com/  
HSU-ANT/beaqlejs](https://github.com/HSU-ANT/beaqlejs)

- code and demos are available on GitHub
- GPLv3 licence
- feedback and contributions are appreciated!