# EVALUATION OF WEBRTC AS A FRAMEWORK FOR VOICE RECORDINGS IN ONLINE SURVEYS

*Anabell Hacker*[1,2]*, Iris Sidonie Bakker*[2]*, Ingo Siegert*[1,2]

[1]*Mobile Dialog Systems, Otto von Guericke University Magdeburg*
[2] *Research Group on Intelligent Assistive Systems for Psychotherapy, University Hospital Magdeburg*
*{anabell.hacker, ingo.siegert}@ovgu.de*

**Abstract:** Web-based speech data collection is increasingly important for large-scale linguistic and phonetic research. WebRTC enables browser-based audio recording without native apps, potentially lowering participation barriers. This study compares three WebRTC-based recording frameworks, native WebRTC, RecordRTC, and MediaStream Recorder, integrated into the online survey platform SoSci Survey. Speech data were collected under varying recording settings and across different devices, operating systems, and browsers. Recordings from 80 participants were manually analysed for technical artefacts. All frameworks showed similar artefact types, including clipping, noise, echo, and amplitude distortions, which occurred more frequently in sustained vowels than in read speech. Artefacts were most prevalent on Android and Windows systems using Chrome, while Apple devices showed fewer distortions. Avoiding enforced audio formats reduced clicking artefacts on some platforms.

## 1 Introduction

The fast and easy collection of high-quality speech data has a central challenge in many areas of speech technology, including automatic speech recognition, voice print analysis, and clinical assessment [1]. Large-scale speech corpora are essential for studying phonetic variation, speech production and perception, as well as for developing and evaluating speech-based technologies. Traditionally, speech data are collected under controlled laboratory conditions to ensure consistent and high-fidelity recordings. However, lab-based recordings are time-consuming, costly, and may reduce the naturalness of speech due to the unfamiliar environment for participants [2].

To address these limitations, an increasing number of speech corpora are collected outside the laboratory using native mobile applications [3, 4]. Such approaches enable recordings in more natural environments and facilitate participation by individuals who might otherwise be excluded due to travel distance, health constraints, or limited access to university facilities. Nevertheless, increased accessibility does not automatically translate into higher participation rates. Some participants are reluctant to install unfamiliar applications on their personal devices [5], motivating the growing adoption of browser-based recording solutions.

Browser-based approaches allow participants to contribute speech data using their own devices without installing additional software, thereby lowering technical barriers and potentially increasing participation and demographic diversity. From a researcher's perspective, they also reduce the effort associated with developing and maintaining native applications across multiple mobile ecosystems, adapting to frequent operating system updates, and navigating app store certification and review processes.

Among the browser-based solutions, WebRTC (Web Real-Time Communication) [6] has emerged as a de facto standard for real-time audio and video communication in modern browsers. It provides direct access to audio input devices and serves as the foundation for many speech recording tools. JavaScript libraries such as RecordRTC [7] extend WebRTC for media recording, while the MediaStream Recorder API [8] offers a native browser interface for capturing audio streams without external dependencies. Integrating these technologies into online survey platforms, such as SoSci Survey [9], enables large-scale, web-based speech data collection.

However, in a remotely recorded speech corpus collected via a WebRTC-based integration in SoSci Survey, we observed recurring acoustic artefacts, including clicking noises and abrupt amplitude changes. Detailed auditory and visual inspection suggested that technical aspects of browser-mediated audio capture—such as echo cancellation, noise suppression, automatic gain control, and codec negotiation—may adversely affect signal fidelity. This motivates a systematic comparison of the audio quality produced by different recording frameworks. Accordingly, this study evaluates recordings obtained using native WebRTC, RecordRTC, and MediaStream Recorder, all implemented within the same online questionnaire environment.

While prior work has investigated codec behaviour and perceived speech quality in networked audio systems [10, 11, 12, 13], the recording layer of browser-based speech data collection remains underexplored. In particular, there is a lack of systematic, framework-level comparisons that evaluate how different WebRTC-based recording implementations handle critical signal processing mechanisms such as echo cancellation, noise suppression, and automatic gain control. As a result, it remains unclear to what extent observed artefacts originate from codec limitations, browser behaviour, or framework-specific implementations. Addressing this gap is essential for assessing the reliability and reproducibility of remotely collected speech corpora.

## 2 Methods

Following the observation of recurring acoustic artefacts in speech data collected remotely via SoSci Survey, this study systematically investigates the browser-based recording frameworks involved. The goal is to reproduce the observed artefacts under controlled conditions, identify their technical causes, and assess their impact on recording quality.

An experimental research design was employed to compare the audio quality of voice recordings obtained using three browser-based recording frameworks: native WebRTC, RecordRTC, and the MediaStream Recorder API. All recordings were collected remotely using the SoSci Survey platform[1]. Recording conditions were systematically varied with respect to framework settings and technical parameters, including browser type, operating system, and recording hardware.

The collected audio samples were analysed both auditorily and visually using Praat and Audacity and manually classified according to predefined artefact categories (e.g., clipping, clicking noises, abrupt amplitude changes, echo). The study follows a quantitative approach aimed at identifying technical factors that affect the reliability and signal fidelity of browser-based speech recordings in online experiments.

[1]To exclude the possibility that SoSci Survey itself caused the artefacts, the same HTML and JavaScript code was tested on a local website. The locally recorded audio files exhibited the same artefacts as those collected via SoSci Survey, indicating that the artefacts originate from the browser-based recording frameworks rather than from the survey platform.

### 2.1 Online Survey Design

Three separate online surveys were created in SoSci Survey, one for each recording framework. Participants first provided demographic information and detailed metadata about their recording setup, including device type (e.g., mobile phone or computer), device model, operating system, microphone hardware (built-in or external), browser, and recording environment (e.g., home/office or public space). In addition, technical parameters such as manufacturer, operating system, browser identification, browser type, and device format were automatically logged by the survey system.

Participants then recorded two types of speech material: sustained vowels (/a:/ and /i:/) and a read-speech passage of the German version of *The North Wind and the Sun*. The sustained vowels were recorded multiple times under different predefined framework settings, while read speech was recorded once per participant and framework. During recording, participants were shown a live intensity display to help avoid clipping, and they were able to listen to each recording immediately afterwards to check for obvious errors.

### 2.2 Framework Settings

The recordings were implemented using the *upload media files* question type in SoSci Survey, which embedded customised HTML and JavaScript code to initialise and configure the respective recording frameworks. For native WebRTC and MediaStream Recorder, the activation states of noise suppression, echo cancellation, and automatic gain control were systematically varied to assess their influence on recording quality. In contrast, RecordRTC offers only limited control over audio processing and provides a basic echo-cancellation fix for Microsoft Edge without fine-grained parameter configuration.

Read-speech recordings were collected using the default settings of WebRTC and MediaStream Recorder. Sustained vowels were recorded in multiple rounds with all three audio processing options disabled, all enabled, and each option enabled individually while the others remained disabled. As native WebRTC allows more detailed audio control, additional vowel recordings were made with an enforced sample rate of 44.1 kHz, a sample size of 16 bit, a limiter/compressor to reduce clipping, and a reduced gain value of 0.9. Since RecordRTC does not support these settings, only a single vowel recording and one read-speech recording were collected in that condition. All code used in the surveys is publicly available at `github.com/Mobile-Dialog-Systeme/webrtc-checker`.

## 3 Results

### 3.1 Sample

A total of 80 participants (42 female, 38 male) contributed speech recordings, yielding 57 read-speech samples and 338 sustained vowel recordings. The mean participant age was 36 years (SD = 9.4).

The native WebRTC survey was completed by 36 participants (14 f, 22 m; mean age 35 years, SD = 6.0), resulting in 20 read-speech recordings and multiple vowel recordings per participant depending on framework settings. Using RecordRTC, 18 participants (12 f, 6 m; mean age 34 years, SD = 3.7) contributed 17 read-speech recordings and 18 vowel recordings. The MediaStream Recorder survey was completed by 26 participants (16 f, 10 m; mean age 41 years, SD = 16.6), yielding 20 read-speech recordings and between 18 and 26 vowel samples per framework setting.

Differences in the number of recordings resulted from participant dropout. However, as

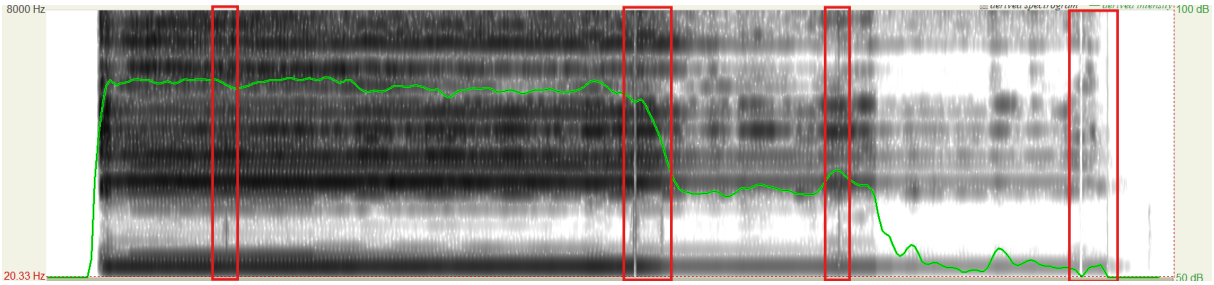



**Figure 1** – Sustained /iː/ vowel showing abrupt amplitude jumps and clicking artefacts (highlighted in red; Praat spectrogram with intensity contour).

recordings were saved upon completion of each survey page and informed consent had been given beforehand, all available recordings were included in the analysis.

Most recordings were made on smartphones ($n = 60$), with fewer contributions from laptops or desktop computers ($n = 20$). Android ($n = 34$) and iOS ($n = 21$) were the most common operating systems, followed by Windows ($n = 20$). Browsers included Chrome ($n = 22$), Safari ($n = 22$), Firefox ($n = 16$), Samsung Internet ($n = 17$), Edge ($n = 2$), and Ecosia ($n = 1$). The majority of recordings were made at home ($n = 74$) using built-in microphones ($n = 68$), with limited use of Bluetooth headsets ($n = 11$) and external microphones ($n = 1$). Overall, the sample reflects a wide range of realistic recording environments.

### 3.2 Evaluation of WebRTC Frameworks

All recordings were analysed auditorily and visually using Praat and Audacity to identify and classify technical artefacts. Across all three frameworks, seven artefact types were identified: clipping, clicking or popping noises, abrupt amplitude jumps (Fig. 1), broadband noise, echo, very low amplitude levels, and sudden quality degradation. Artefacts occurred more frequently in sustained vowels than in read speech.

Artefact counts were normalised by the number of recordings per condition (e.g., browser, operating system, device type) to account for unequal sample sizes across frameworks and settings. Browser- and operating-system–specific distributions are summarised in Figures 2 (a) and (b).

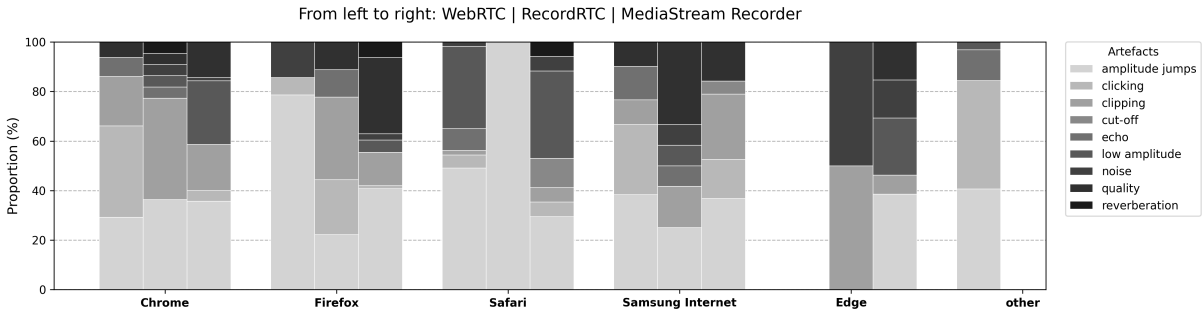
Across all **WebRTC** recordings, amplitude jumps were the most frequent artefact (40.5%), followed by clicking or popping noises (26.5%), echo (10.7%), and clipping (9.1%). Broadband noise was rare (1.0%). Artefacts were predominantly observed in smartphone recordings (94.5%), while laptop and desktop recordings showed fewer and less diverse degradations.

Strong browser- and platform-specific patterns emerged. Chrome recordings combined clicking, amplitude jumps, and clipping, whereas Firefox and Windows environments were dominated by amplitude jumps. Safari recordings showed a comparatively high proportion of low-amplitude artefacts. Android devices exhibited the most heterogeneous artefact profiles, while iOS recordings were largely characterised by amplitude jumps and low signal levels.

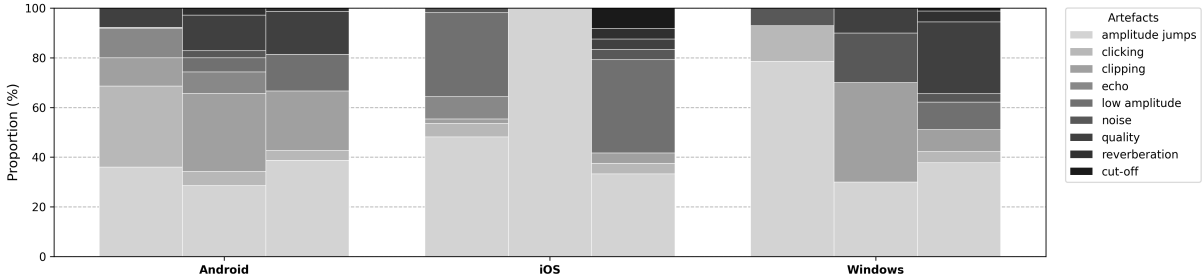
Varying WebRTC processing pipelines influenced artefact distributions but did not eliminate dominant degradation patterns. Across configurations, amplitude instability and clicking artefacts remained prevalent.

**RecordRTC** recordings were mainly affected by dynamic artefacts, with clipping (32.6%) and amplitude jumps (30.4%) being most frequent. Clicking artefacts were comparatively rare (4.3%). Browser and platform effects were pronounced, with Chrome recordings dominated by clipping and amplitude jumps, while Firefox and Samsung Internet exhibited more heterogeneous degradation profiles. All iOS-based recordings showed amplitude jumps exclusively.

Device type had only minor influence, whereas microphone choice affected artefact composition. Built-in microphones showed mixed degradation patterns, while Bluetooth headsets




**(a)** Browser-based artefact distribution




**(b)** Operating system–based artefact distribution

**Figure 2** – Distribution of audio artefact types for native WebRTC, RecordRTC, and MediaStream Recorder. Browser-specific effects are shown in (a), operating system–specific effects in (b).

were primarily affected by amplitude instability and quality-related artefacts.

**MediaStream Recorder** recordings were dominated by amplitude jumps (37.5%), followed by quality-related distortions (20.0%), clipping (15.5%), and low-amplitude artefacts (15.5%). Browser and operating system effects were clearly differentiated, with Safari recordings particularly affected by low signal levels and Chrome recordings showing combined amplitude and clipping artefacts.

Framework-level processing configurations substantially influenced artefact composition. Different settings emphasised distinct degradation patterns, indicating a high sensitivity of MediaStream Recorder to processing choices.

**Across all frameworks**, dynamic and level-related artefacts—particularly amplitude jumps and clipping—were most prevalent. Native WebRTC exhibited strong browser- and device-specific instability, with frequent clicking and echo artefacts. RecordRTC reduced clicking artefacts but remained susceptible to clipping and amplitude instability, especially on mobile platforms. MediaStream Recorder showed the most systematic dependence on framework-level processing choices, yielding more predictable artefact patterns when processing was minimised.

# 4 Artefact Detection Tool

To enable rapid artefact detection in audio recordings, a Python-based detection tool was developed. The tool is specifically designed for vowel recordings, as amplitude changes can be most reliably identified in signals with near-constant intensity. In addition, vowel production provides a comparatively controlled speech condition, allowing unintentional recording artefacts to be more clearly distinguished from speaker-specific characteristics such as habitual loudness, articulation style, or prosodic variation. Audio analysis is performed using the Python library librosa, which provides functionality for framing, spectral feature extraction, and the computation of root mean square (RMS) energy on a frame-wise basis.

The detection tool supports the identification of strong background noise, clipping, and clicking artefacts. Clicking artefacts are further differentiated into those caused by abrupt

sample-to-sample discontinuities and those resulting from short-term amplitude suppression. In addition, the tool issues warnings for recordings containing no signal or exhibiting very low overall volume.

Given their frequent occurrence, a primary objective of the tool is the detection of abnormal amplitude changes. A key challenge lies in distinguishing artefactual amplitude variations from naturally occurring fluctuations. This distinction is particularly difficult because amplitude changes may occur very rapidly (within less than one second) or evolve more gradually over a period of two to five seconds, resembling natural volume dynamics. Moreover, amplitude reductions can lead to signal levels that are barely distinguishable from background noise.

Additional objectives of the tool include improved sensitivity to amplitude reductions resulting in very low signal levels and the detection of reverberation effects in audio recordings. The current version of the tool is publicly available on GitHub: `https://github.com/Mobile-Dialog-Systeme/webrtc-checker`.

## 5 Discussion and Outlook

This study demonstrates that browser-based speech recording is not a neutral acquisition step: across all three frameworks, the recording pipeline systematically shaped signal quality and introduced characteristic artefact profiles. Dynamic and level-related degradations—in particular abrupt amplitude jumps and clipping—were prevalent across conditions, suggesting that client-side capture and processing (including device-dependent gain staging and browser audio pipelines) can dominate the resulting signal characteristics in remote data collection.

A consistent finding was the markedly lower incidence of clicking artefacts in RecordRTC and MediaStream Recorder compared to the native WebRTC implementation. While this pattern is compatible with differences in encoding and buffering strategies across implementations, the present design does not isolate codec, container, and browser-internal processing effects. Future work should therefore explicitly fix and log codec, sampling rate, and bitrate settings (where possible) to disentangle encoding-related artefacts from capture- and processing-induced distortions.

Importantly, MediaStream Recorder provided the most controllable artefact structure: disabling noise suppression, echo cancellation, and automatic gain control yielded the most predictable degradation profile across heterogeneous user devices. From a methodological perspective, prioritising the avoidance of irreparable distortions is crucial. Clipping and transient discontinuities permanently compromise spectral and temporal properties and are difficult to correct reliably post hoc. In contrast, low-amplitude recordings can sometimes be mitigated by careful post-processing (e.g., normalisation), although such procedures may also amplify background noise and should be applied cautiously.

Taken together, none of the evaluated solutions fully eliminated artefacts under real-world conditions. However, within the tested setup, MediaStream Recorder with all processing modules disabled emerged as the most suitable option for remote online speech recordings when the goal is a stable and analytically tractable signal profile.

While the present study provides a systematic comparison of browser-based recording frameworks under realistic conditions, certain constraints should be considered when interpreting the results. First, conditions were not perfectly balanced across browsers, operating systems, and devices, and some categories were represented by only few recordings, limiting fine-grained subgroup analyses. Second, artefact labels were based on auditory and visual inspection; while this procedure is standard in exploratory quality analyses, it introduces a degree of subjectivity and motivates future validation with automated detectors and inter-rater agreement. Finally, the results reflect the browser and operating-system versions available at the time

of data collection; given frequent updates to WebRTC stacks, the observed artefact profiles may evolve over time.

Future work should (i) extend the evaluation to larger and more balanced samples per browser–OS combination, (ii) systematically test fixed codecs and sampling configurations to isolate encoding effects, and (iii) integrate automated quality screening into the collection pipeline. The provided artefact detection tool is a first step towards scalable quality control. A built-in robust SNR estimation, reverberation metrics, and automated flagging rules that trigger re-recording prompts during data collection would be really helpful for online audio recordings.

## 6 Acknowledgements

This research was funded by the BMFTR within the project Medinym (KI-basierte Anonymisierung personenbezogener Patientendaten in klinischen Text- und Sprachdatenbeständen), funded by the European Union – NextGenerationEU.

The authors would like to thank the anonymous speakers who have contributed their voices to this experiment.

## References

[1] CHEN, S., C. WANG ET AL.: *WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing*. *arXiv preprint arXiv:2110.13900*, 2021. URL https://arxiv.org/abs/2110.13900.

[2] SIEGERT, I.: *"Alexa in the wild" – Collecting Unconstrained Conversations with a Modern Voice Assistant in a Public Environment*. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pp. 608–612. European Language Resources Association, Marseille, France, 2020. URL https://aclanthology.org/2020.lrec-1.77/.

[3] WEIRICH, M., D. DURAN, and S. JANNEDY: *Gender and age based $f_0$-variation in the German Plapper Corpus*. In *Proceedings of Interspeech 2024*, pp. 1565–1569. 2024. doi:10.21437/Interspeech.2024-1592.

[4] LEEMANN, A., M.-J. KOLLY, R. PURVES, D. BRITAIN, and E. GLASER: *Crowdsourcing Language Change with Smartphone Applications*. *PLOS ONE*, 11(1), p. e0143060, 2016. doi:10.1371/journal.pone.0143060.

[5] WENZ, A. and F. KEUSCH: *Increasing the Acceptance of Smartphone-Based Data Collection*. *Public Opinion Quarterly*, 87(2), pp. 357–388, 2023. doi:10.1093/poq/nfad019. https://academic.oup.com/poq/article-pdf/87/2/357/50873578/nfad019_supplementary_data.pdf.

[6] THE WEBRTC PROJECT AUTHORS: *WebRTC – Real-Time Communication for the Web*. 2011. URL https://webrtc.org/. Accessed: Oct 27, 2025.

[7] KHAN, M.: *RecordRTC. Version 5.6.2*. 2020. URL https://recordrtc.org/. Accessed: Oct 27, 2025.

[8] THE WEBRTC PROJECT AUTHORS: *MediaStream Recorder – W3C Working Draft*. 2025. URL https://www.w3.org/TR/mediastream-recording/. Accessed: Nov 06, 2025.

[9] LEINER, D. J.: *SoSci Survey. Version 3.7.06*. 2025. URL https://www.soscisurvey.de/. Accessed: Oct 27, 2025.

[10] TROJAHN, F., M. MESZAROS, M. MARUSCHKE, and O. JOKISCH: *Surround Sound Processed by Opus Codec: A Perceptual Quality Assessment*. In *Elektronische Sprachsignalverarbeitung (ESSV) 2017*, pp. 300–307. 2017. URL https://www.essv.de/pdf/2017_300_307.pdf.

[11] JOKISCH, O., M. MARUSCHKE, M. MESZAROS, and V. IAROSHENKO: *Audio and Speech Quality Survey of the Opus Codec in Web Real-Time Communication*. In *Elektronische Sprachsignalverarbeitung (ESSV) 2016*, pp. 300–307. 2016. URL https://www.essv.de/pdf/2016_254_262.pdf.

[12] SIEGERT, I., O. JOKISCH, A. F. LOTZ, F. TROJAHN, M. MESZAROS, and M. MARUSCHKE: *Acoustic Cues for the Perceptual Assessment of Surround Sound*. In A. KARPOV, R. POTAPOVA, and I. MPORAS (eds.), *Speech and Computer*, pp. 65–75. Springer International Publishing, Cham, 2017.

[13] LÖSCH, E., A. ZIMMERMANN, A. SCHENK, and O. JOKISCH: *Entwicklung einer universellen Audio- und Datenschnittstelle zur Sprachqualitätsmessung in digitalen Funknetzen*. In *Elektronische Sprachsignalverarbeitung (ESSV) 2016*, pp. 254–262. 2016. URL https://www.essv.de/pdf/2016_246_253.pdf.