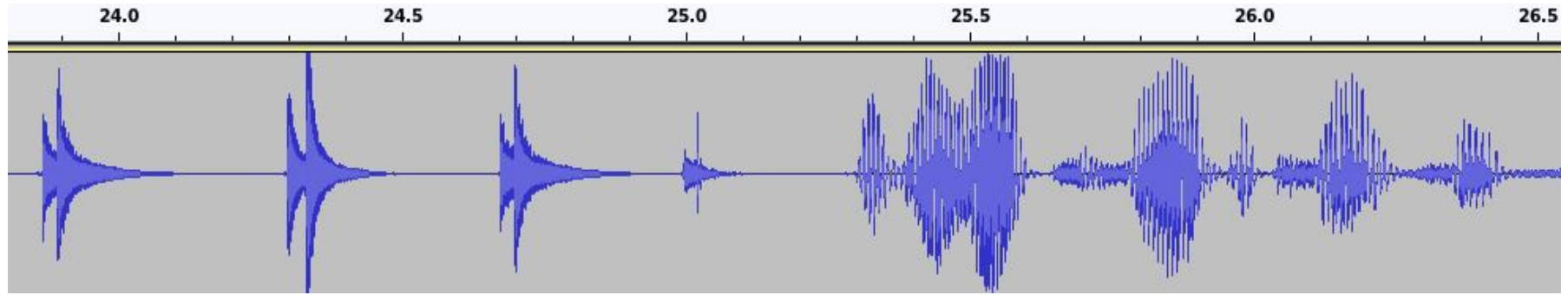


1. This TuneUrl-POC project is using [AudioContext](#) for playing audio files. The main JavaScript is at `src/main/webapp/js/audio-demo.js`

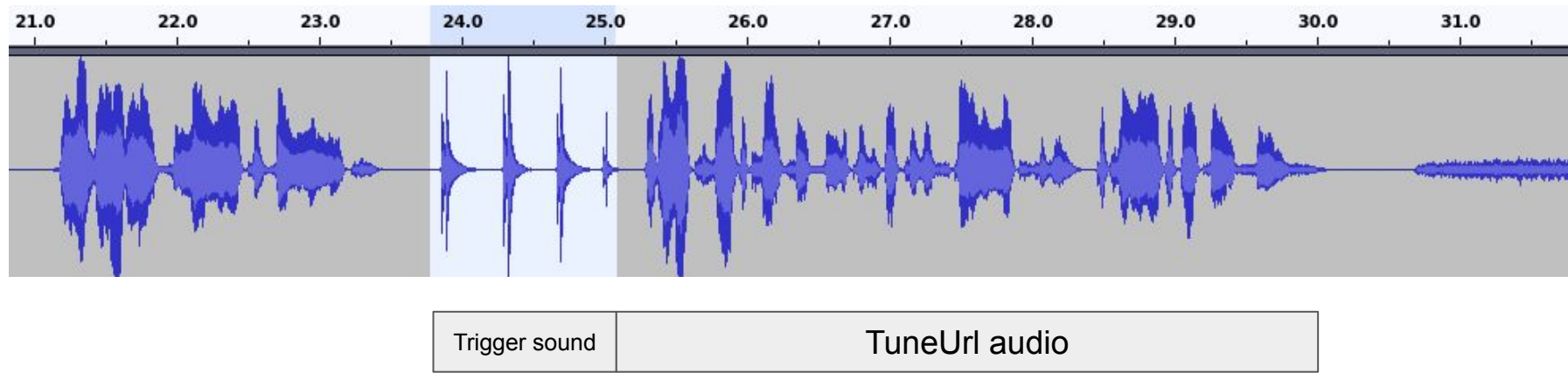
2. The audio stream to play is defined at `LOAD_FROM_THIS_URL`. Since the [AudioContext](#) requires a loadable file, the audio stream is converted to an audio file by calling `POST /dev/v3/saveAudioStream`

3. The audio file is then scan for trigger sound to locate TuneUrl embedded into the audio stream.



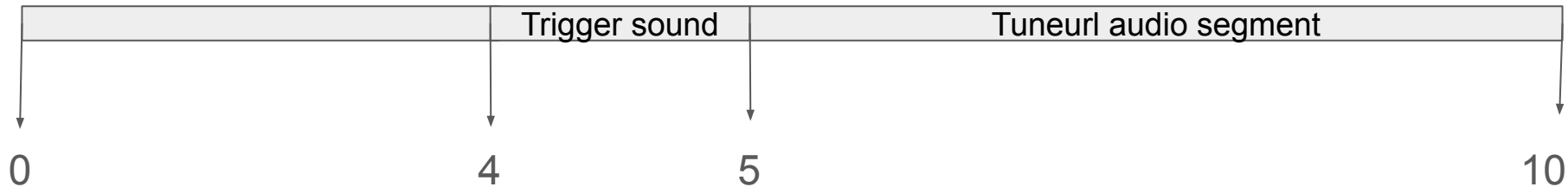
The sample audio above is showing the trigger sound with 1000 milliseconds duration. It is use to mark where the TuneUrl is located in the audio file. The TuneUrl audio segment is an audio after the trigger sound.

In order to detect the trigger sound in an audio stream, the application is using the audio identification method known as audio fingerprint. Audio fingerprinting is compute-intensive process and currently implemented on the server. The POST `/dev/v3/calculateFingerprint` endpoint is use for calculating the audio segment fingerprint while POST `/dev/v3/evaluateOneSecondAudioStream` endpoint is use for comparing two fingerprints.



The trigger sound is an audio segment is currently have a 1000 milliseconds duration while the TuneUrl audio is a least 5000 milliseconds duration.

In order to detect the trigger sound every 100 milliseconds interval, I need to process a 10 seconds audio segment and identify the trigger sound in less than 5 seconds. During initialization, I have made a 10 seconds delay before showing the play button. This enables the scanning of trigger sound to at least be 10 seconds ahead of the playing of the audio.



From 0 to 5 seconds, the trigger sound fingerprint is scanned by comparing it with the fingerprints of the 10 seconds audio segment. If the trigger sound is on the 10 seconds audio segment, the 5 seconds audio segment after the trigger sound is converted into a fingerprint for later call to a matching API [here](#).

Fingerprint scanning is done every 100 milliseconds interval. For each 100 milliseconds, five fingerprints are created and compared with the trigger sound fingerprints. See `com.tuneurl.webrtc.util.controller.OneSecondAudioStreamController#evaluateOneSecondAudioStream()` for details

```

async function startCanvas() {
  console.log('startCanvas');
  initVariables();
  displaySpinner(true);
  initChannelData(0);
  setButtonPlayOrPause(true);
  await doLogin();
  if (hasJWT()) {
    await initTriggerAudio(TRIGGERSOUND_AUDIO_URL);
    if (triggerFingerprintSize > 0) {
      await initLoadFromUrl(Load_From_This_URL);
      if (IF_LOAD_ALL_AUDIO_STREAM) {
        emitTuneUrlInstruction(true, 0);
        await saveAudioEx(initAllTagsEx);
      } else {
        saveAudio();
      }
    }
  }
  displaySpinner(false);
}

```

The JavaScript main process started at function startCanvas().