Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

ИИ для звука: чистка, разделение дорожек, генерация музыки

Наш конвейер: Auphonic API, Demucs v4 и UVR локально, Lyria на Vertex AI

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Записи планёрок, вебинаров и роликов приходят с гарнитур и телефонов: разные уровни у спикеров, эхо кабинета, гул вентиляции, музыка на фоне. Ручная доводка съедает час на выпуск, без неё материал звучит тише конкурентов и его не досматривают. Мы закрываем это конвейером: чистка и нормализация по LUFS, локальное разделение дорожек, генерация фона с фиксацией лицензии.

Почему это важно бизнесу

- Час звукорежиссёра на каждый выпуск превращается в 3-5 минут машинного времени и очередь заданий
- Тихий или скачущий по громкости ролик теряет досмотры: площадка сама подтянет уровень и сломает динамику
- Записи планёрок и переговоров с персональными данными нельзя выгружать в чужое облако
- Чужая фоновая музыка в соцсетях — это блокировка ролика и претензии правообладателя
- Без реестра промптов, seed и лицензий контент невозможно переделать и защитить

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

-16 LUFS

loudnesstarget: значение по умолчанию в Auphonic и наш профиль для речи и подкастов

Auphonic API: algorithms.json

-1.0 dBTP

maxpeak: дефолт 99.0 лимитер не включает, мы ставим запас на кодирование в AAC

Auphonic API: algorithms.json

12 дБ

denoiseamount для офисных записей; шкала 3-100 дБ, выше 18 на речи слышен металл

Наш пресет чистки речи

0.25

--overlap в Demucs v4 по умолчанию: меньше — быстрее, но слышны швы сегментов

Demucs README: параметры разделения

32.8 с

предельная длина клипа lyria-002; длинный фон собираем склейкой с кроссфейдом

Vertex AI: спецификации Lyria 2

4 стема

htdemucs: vocals/drums/bass/other, WAV 44.1 кГц стерео; 6 стемов — htdemucs_6s

Demucs README: модели и выход

РЕАЛИЗАЦИЯ 1 ИЗ 3

Автоматическая чистка речи: Auphonic API и контроль громкости

Что настраиваем

Watch-папка на нашем Linux-узле обработки: приём с файлового сервера клиента, выдача мастеров в шару маркетинга

Как мы это делаем

1. Исходник падает в SMB-шару in/; systemd path unit постит файл на https://auphonic.com/api/simple/productions.json с preset=UUID и action=start
2. Авторизация: Authorization: bearer <api_key>. В пресете denoise=true, denoisemethod=dynamic, denoiseamount=12, levelerstrength=100, filtering=true, gate=true
3. Громкость: normloudness=true, loudnesstarget=-16, maxpeak=-1.0, maxlra=0 (авто); второй пресет с loudnesstarget=-14 под стриминговые площадки
4. Готовность ловим на webhook-URL из запроса, мастер кладём в out/ и меряем ffmpeg -af loudnorm=print_format=json: расхождение I больше 1 LU — ретрай
5. Офлайн-контур без облака: ffmpeg в два прохода — highpass=f=80 и afftdn, затем loudnorm с measured_I/TP/LRA и linear=true

РЕЗУЛЬТАТ

Выпуск подкаста или вебинара доходит от сырой записи до готового мастера за 3-5 минут без участия человека: уровни спикеров выровнены, громкость предсказуемо совпадает с требованиями площадки, и площадка больше не давит трек своим нормализатором.

КЛЮЧЕВОЙ НЮАНС

Однопроходный loudnorm работает как динамический компрессор и даёт помпаж. Мы делаем два прохода с measured_I/TP/LRA и linear=true, а пики ограничиваем явно: дефолт maxpeak=99.0 лимитер не включает.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Локальное разделение дорожек: Demucs v4 и UVR без облака

Что настраиваем

Изолированный узел: GPU от 3 ГБ VRAM (на параметрах по умолчанию нужно ~7 ГБ) или CPU, только SMB in/out, без выхода в интернет

Как мы это делаем

1. Ставим demucs в отдельный venv на Python 3.11, тянем веса htdemucs_ft: он в 4 раза медленнее htdemucs, но точнее; выход — WAV 44.1 кГц стерео
2. Типовой запуск: demucs -n htdemucs_ft --two-stems=vocals --shifts=1 --overlap=0.25 -o /srv/stems input.wav — на выходе vocals.wav и no_vocals.wav
3. Мало VRAM — уменьшаем --segment (Hybrid Transformer не принимает больше 7.8 с); на CPU (-d cpu) закладываем время x10 и гоним батч по systemd timer
4. Ручные кейсы ведём в UVR GUI 5.6: у MDX23C и MDX-Net крутим Segment Size и Overlap, а Aggression Setting есть только у моделей VR Architecture
5. Retention: промежуточные стемы живут 14 дней, мастера уходят в бэкап, логи заданий пишем в syslog узла

РЕЗУЛЬТАТ

Речь спикера отделяется от фоновой музыки и посторонних дорожек прямо в периметре клиента: файл не покидает контур, а чистый вокал-стем сразу идёт на расшифровку и субтитры без ручной переработки монтажёром.

КЛЮЧЕВОЙ НЮАНС

Модели разделения обучены на музыке, а не на речи в комнате: гул и реверберацию они не убирают, а размазывают фазой. Порядок операций у нас жёсткий — сначала разделение, потом денойзер и нормализация.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Фоновая музыка: Lyria на Vertex AI и учёт лицензий

Что настраиваем

Проект GCP маркетинга клиента: сервис-аккаунт с ролью roles/aiplatform.user, раннер очереди заданий вне периметра

Как мы это делаем

1. Включаем Vertex AI API и шлём POST на /v1/projects/PROJECT/locations/global/publishers/google/models/lyria-002:pre ключ — в хранилище секретов
2. В instances задаём prompt (только US English) и negative_prompt; seed и sample_count взаимоисключающие — либо повторяемость, либо до 4 клипов
3. Ответ — audioContent в base64, WAV 48 кГц; квоту 10 запросов в минуту на модель и нестабильный доступ из РФ закрываем внешним раннером с очередью
4. Трек нужной длины собираем из 3-4 клипов с одним seed: ffmpeg acrossfade на 1.5 с, готовую подложку нормализуем к -20 LUFS
5. Своего водяного знака у lyria-002 нет (audio watermarking и C2PA не поддержаны) — маркируем сами: реестр промпт, seed, модель, дата, тариф

РЕЗУЛЬТАТ

Маркетинг получает подложку под ролик за минуты и без риска претензий правообладателя, а реестр промптов и seed позволяет через полгода воспроизвести ту же музыкальную тему для новой серии видео в том же стиле.

КЛЮЧЕВОЙ НЮАНС

Право на коммерческое использование даёт тариф конкретного сервиса, а не сам факт генерации. Мы фиксируем тариф и дату генерации в реестре и не берём треки с бесплатных персональных планов в рекламу.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **Одиночный проход loudnorm**
  В один проход фильтр работает динамически и качает громкость. Делаем измерение, затем второй проход с measured_* и linear=true

- ✕ **Денойзер выкручен в максимум**
  denoiseamount 30-100 съедает согласные и даёт металлический тембр. Для речи в кабинете держим 12-18 дБ и слушаем результат

- ✕ **Пиковый лимитер не включён**
  Дефолт maxpeak=99.0 означает отсутствие ограничения: после кодирования в AAC вылезают интерсемпловые пики. Ставим -1.0 dBTP

- ✕ **Один мастер под все площадки**
  Речь мы отдаём в -16 LUFS, стриминговую музыку в -14. Единый файл либо звучит тише, либо площадка сама режет динамику

- ✕ **Разделение вместо шумоподавления**
  Demucs и UVR решают задачу музыкальных стемов. Эхо и гул кондиционера убирает денойзер и дереверб, а не сплиттер

- ✕ **Конфиденциальная запись в SaaS**
  Планёрки, переговоры и медданные не уходят в облако. Для них — локальный узел без интернета и очистка стемов по расписанию

- ✕ **Нет реестра промптов и seed**
  Без сохранённых prompt, seed и версии модели повторить или продолжить музыкальную тему невозможно, ролик придётся переозвучивать

- ✕ **Сырьё пишется в lossy**
  Запись сразу в mp3 96 кбит/с убивает запас для обработки. Пишем WAV 48 кГц и отдельную дорожку на каждого спикера

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Один пресет чистки на компанию: -16 LUFS, maxpeak -1.0, denoiseamount 12
- Запись в WAV 48 кГц, петличка или USB-микрофон, дорожка на спикера
- Проверка мастера через loudnorm print_format=json перед публикацией
- Правило: конфиденциальные записи в облачные сервисы не загружаем

НОРМАЛЬНО

- Watch-папка in/out на файловом сервере, задания и webhook готовности
- Два пресета: -16 LUFS для речи и -14 LUFS для музыки на стриминге
- Локальный узел Demucs v4 (htdemucs_ft) для записей внутри периметра
- Реестр треков: промпт, seed, модель, тариф, дата — в шаре маркетинга

ХОРОШО

- Автоконтроль после рендера: ретрай при отклонении I больше 1 LU
- GPU-узел от 8 ГБ VRAM, ночные батчи разделения по systemd timer
- Транскрипт и субтитры строим по чистому вокал-стему, поиск по архиву
- Retention 14-30 дней на стемы, мастера и реестр лицензий — в бэкап

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Знаем ли мы целевую громкость каждой площадки в LUFS и записана ли она в пресет?
- [ ] Включён ли ограничитель пиков или maxpeak остался в значении по умолчанию 99.0?
- [ ] Пишем ли мы исходники в WAV и храним ли их до сдачи проекта?
- [ ] Определено ли, какие записи вообще нельзя отправлять в облачный сервис?
- [ ] Есть ли локальный контур обработки на случай недоступности внешнего сервиса?
- [ ] Ведётся ли реестр сгенерированной музыки с промптом, seed и условиями тарифа?
- [ ] Проверяется ли готовый мастер измерением I, TP и LRA, а не только на слух?
- [ ] Настроен ли срок хранения промежуточных стемов и очистка временных папок?
- [ ] Есть ли шаблон записи планёрок: микрофоны, отдельные дорожки, комната?
- [ ] Понятно ли, кто в компании отвечает за выпуск и кто подтверждает лицензии?

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.

ЧТО МЫ ДЕЛАЕМ

Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Собираем конвейер обработки: watch-папка, очередь заданий, пресеты и автопроверка громкости
- Разворачиваем локальный узел Demucs v4 и UVR для записей, которые нельзя отдавать в облако
- Настраиваем генерацию фона на Vertex AI: сервис-аккаунт, квоты, реестр промптов и seed
- Пишем регламент: форматы записи, целевые LUFS, сроки хранения, порядок проверки лицензий
- Обучаем сотрудника-оператора и передаём шаблоны команд ffmpeg под каждую площадку

15+
лет в ИТ-поддержке

50
рабочих мест — наш профиль

МТС
дата-центр, Москва

КОНТАКТЫ

Обсудить вашу задачу

Сайт	**itfresh.ru**
Телефон	**+7 903 729-62-41**
Telegram	**@ITfresh_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.

itfresh.ru

ДОКУМЕНТАЦИЯ И НАШ СТЕК

Техническая база

01 Auphonic Help: Audio Post Production Algorithms (Singletrack) (auphonic.com — 2026)

02 Auphonic API: /api/info/algorithms.json — дефолты и списки значений (auphonic.com — 2026)

03 Auphonic Help: Simple API — productions.json, preset, webhook (auphonic.com — 2026)

04 Demucs README: htdemucs_ft, --shifts, --overlap, --segment (github.com — v4)

05 Ultimate Vocal Remover GUI: релизы, модели MDX23C, настройки (github.com — 5.6)

06 Vertex AI: Lyria 2 (lyria-002) — спецификации и predict (cloud.google.com — 2026)

07 FFmpeg Filters: loudnorm, afftdn, highpass, acrossfade (ffmpeg.org — 8.1)

08 ITU-R BS.1770-5: измерение громкости и true peak (itu.int — 2023)

09 EBU R 128 v4.0: нормализация громкости и максимальный уровень (tech.ebu.ch — 2020)

10 Наш шаблон audio-pipeline: пресеты, watcher, QC-проверка (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.