

# Бот для автоматизации публикации материалов в социальные сети



Доклад подготовил  
Крылышкин Вячеслав

## Проблема:

При регулярной работе с короткими видео автору или контент-менеджеру приходится вручную выполнять повторяющиеся действия: скачать видео, выделить речь, расшифровать содержание, подготовить описание, заголовок, хэштеги и адаптировать текст под разные площадки.

## Идея решения:

Сделать MVP-сервис, который принимает видео в боте (Telegram) и автоматически готовит черновик публикации: транскрипт, текст поста, заголовок, описание и хэштеги.

## Ценность для пользователя

- сокращает ручную работу при подготовке публикации;
- ускоряет обработку видео;
- снижает порог входа: пользователь взаимодействует через Telegram;
- даёт возможность редактировать результат перед публикацией;

# Пользовательский сценарий:

Основной сценарий работы:

1. Пользователь отправляет видео в Telegram-бота.
2. Бот скачивает исходный файл.
3. Из видео извлекается аудиодорожка.
4. Из аудио удаляются паузы и тишина.
5. Очищенное аудио передаётся в модуль транскрибации.
6. На основе текста формируется набор черновиков:
  - а. текст поста VK;
  - б. заголовок;
  - с. описание;
  - д. хэштеги;
  - е. потенциальные тексты под Telegram, Дзен, МАХ, RuTube.
7. Пользователь получает превью.
8. Пользователь может отредактировать отдельные поля.
9. Пользователь выбирает площадку для релиза.
10. В текущем MVP публикация эмулируется, а следующий шаг — подключение реальных API площадок.

# Технические решения

Telegram-bot (aiogram)

Обработка аудио (ffmpeg)

Для удаления пауз -  
silenceremove

Транскрибация (paster-whisper)

Нормализация транскрипта  
(Ollama - qwen2.5:3b)

Генерация превью  
(локальный Stable Diffusion)  
\*демо, работает нестабильно\*

# Интеграция с ODOO

## 1. Хранение задач обработки видео

а. Каждое отправленное видео можно представить как отдельную задачу video.job

## 2. Контроль статусов

Задача проходит состояния:

- draft
- pending\_approval
- approved
- processing
- published
- failed

## 3. Ручное согласование

Пользователь может проверить результат перед публикацией

## 4. Ретраи и ошибки

В модели есть attempt\_count, max\_retries, last\_error.

Это позволяет отслеживать ошибку и повторять обработку

## 5. Публикации по площадкам

Отдельная модель video.publication хранит публикации для Telegram, VK, YouTube, RuTube, Дзен, МАХ

## 6. Медиа-ассеты

Модель video.asset может хранить ссылки на видео, аудио, субтитры, превью-картинки

Спасибо за внимание

