

# **РАСПОЗНАВАНИЕ РЕЧИ НА МОБИЛЬНЫХ УСТРОЙСТВАХ, УПРАВЛЯЕМЫХ LINUX, НА ПРИМЕРЕ ОС АВРОРА**

СПО: ОТ ОБУЧЕНИЯ ДО РАЗРАБОТКИ, 18/06/2021



## ОС АВРОРА

- › Операционная система для мобильных устройств (ARM, x86)
- › POSIX-совместимый дистрибутив Linux, rpm-based, Qt middleware
- › В первую очередь B2B и B2G решения, но и B2C в планах
- › Доверенная среда: ФСТЭК (А4/УД4) и ФСБ (АК2/КС2)
- › Разработка ООО "Открытая мобильная платформа"  
[omp.ru](http://omp.ru)



**ОТКРЫТАЯ  
МОБИЛЬНАЯ  
ПЛАТФОРМА**



# ПРИМЕРЫ РАБОТЫ С РАСПОЗНАВАНИЕМ РЕЧИ

ASR = Automatic Speech Recognition

## Общие случаи

- › Управление мобильным устройством
  - » Переключение режимов работы
  - » Настройка оповещений, уведомлений
  - » Запуск доступных приложений
  - » Вызов телефонных абонетов
  - » И т.п.
- › Распознавание речи (STT = speech to text)
  - » Ведение заметок
  - » Составление коротких сообщений
  - » Перевод речи

## Встретившиеся для ОС Аврора

- › Полевые сотрудники
  - » Получение задач
  - » Подготовка и отправка отчетов
- › Руководители и управляющие
  - » Задачи организации
  - » Работа с почтой, планировщиком, информационными системами

# ПОДХОДЫ К РЕАЛИЗАЦИИ РАСПОЗНАВАНИЯ РЕЧИ

## › Публичные сетевые службы

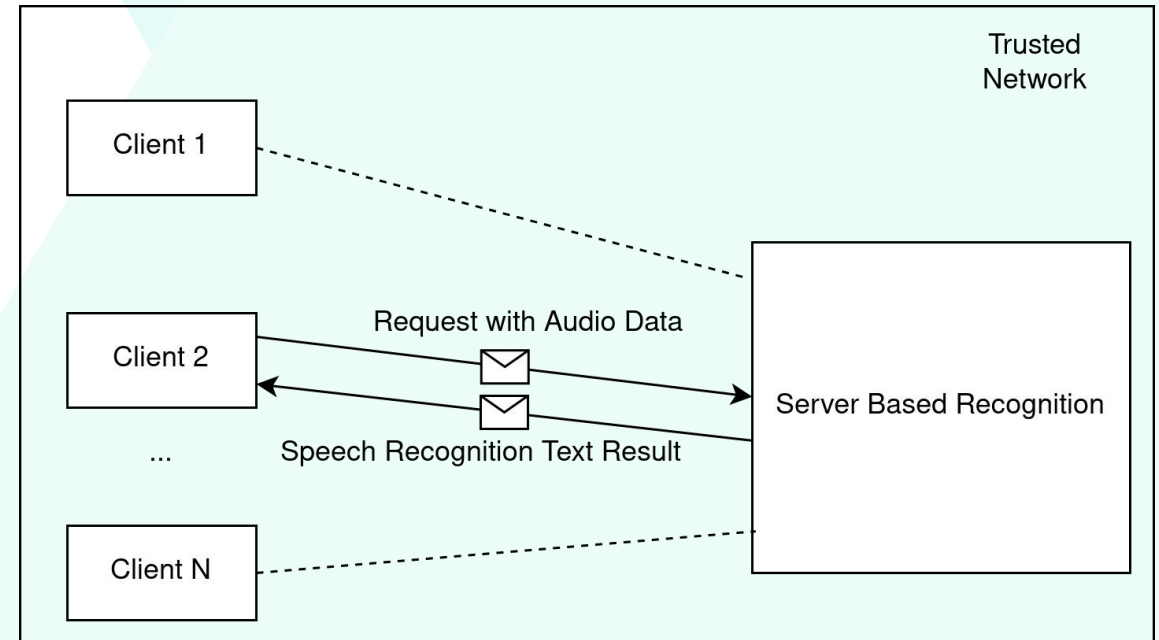
- » Google Cloud Speech-to-Text
- » Apple Speech framework
- » Bing Voice Recognition
- » IBM Speech to Text
- » Яндекс SpeechKit

## › Приватные сетевые службы

- » Могут быть развернуты в доверенной среде заказчика
- » SpeechPro (СТЦ)
- » Jarvisen Translator (IFLYTEK)
- » Cheetah (Picovoice)

## › Распознавание на клиентском устройстве

- » Не требует доступа к сети для каждого запроса
- » Данные на самом устройстве
- » Вычислительные мощности ограничены самим устройством



## **ТРЕБОВАНИЯ ПОЛЬЗОВАТЕЛЯ К РАСПОЗНАВАНИЮ РЕЧИ**

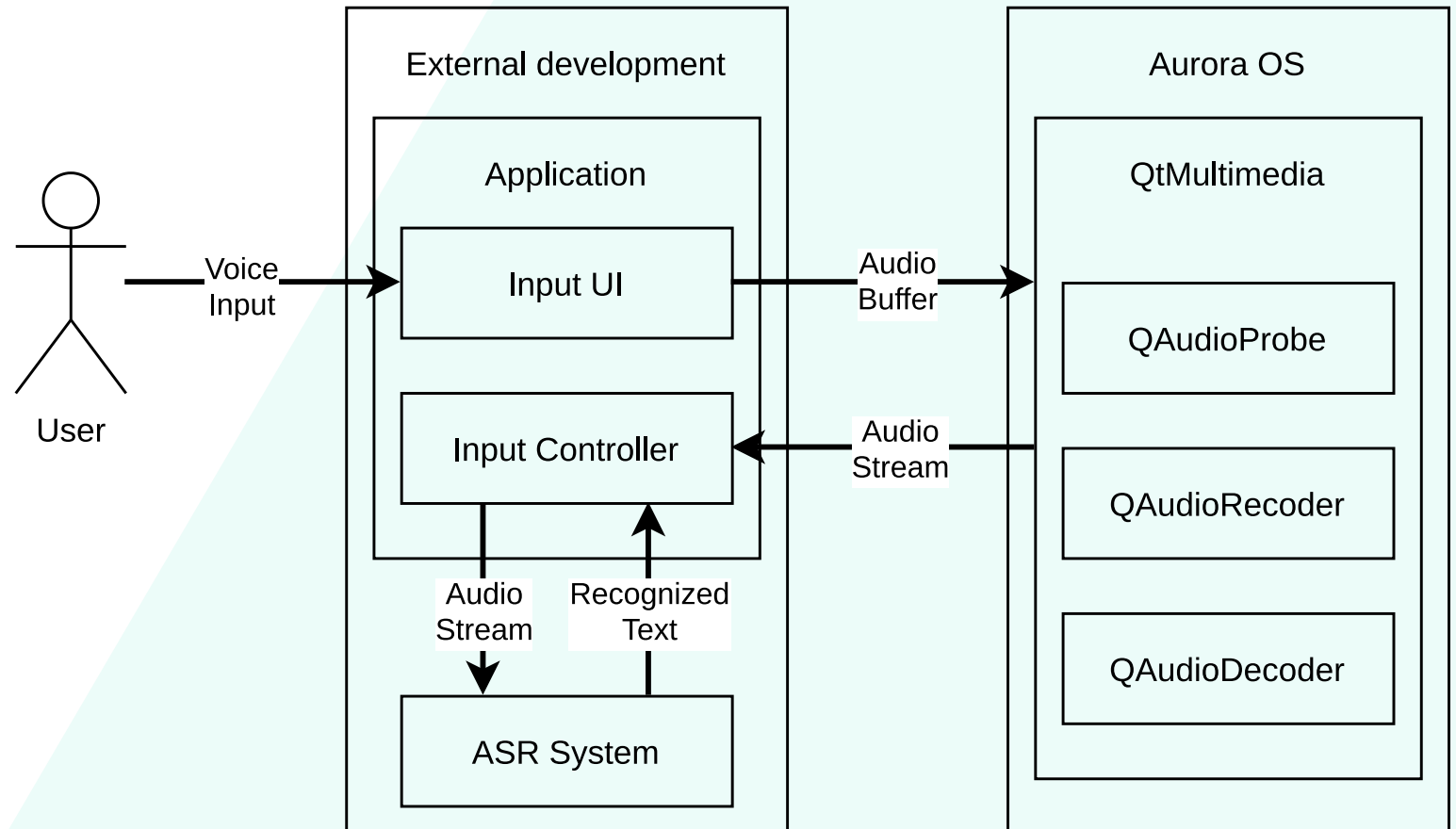
- › Интернационализация и локализация
  - » Русский язык
  - » Английский
- › Распознавание на клиентском устройстве
  - » Доступность при ограничениях сети
  - » Конфиденциальность
- › Скорость распознавания соизмерима или выше сенсорного ввода

## ТЕХНИЧЕСКИЕ ОСОБЕННОСТИ ОС АВРОРА

- › POSIX (подобно Sailfish OS, postmarketOS, Ubuntu Touch, Tizen, и т.п.)
- › make, qmake, cmake для сборки
- › Linaro GNU Toolchain без Gfortran (C++ / gcc)
- › Qt и QML
- › Дополнительные службы для управления устройством
- › Управление пакетами: RPM с дополнительными проверками пакета

# СПОСОБ РЕАЛИЗАЦИИ ASR ДЛЯ ОС АВРОРА

- › Модуль QtMultimedia для захвата звука
  - » QAudioRecorder
  - » QAudioDecoder
  - » QAudioProbe
- › В первую очередь стороне приложение, с планами на системную службу



## **ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ**

- › Возможность сборки для ОС Аврора
- › Свободная лицензия
- › Публичная модель (данные) для русского языка
- › Возможность работы на мобильном железе
- › Активный (не архивный) проект



# ОБЗОР РЕШЕНИЙ

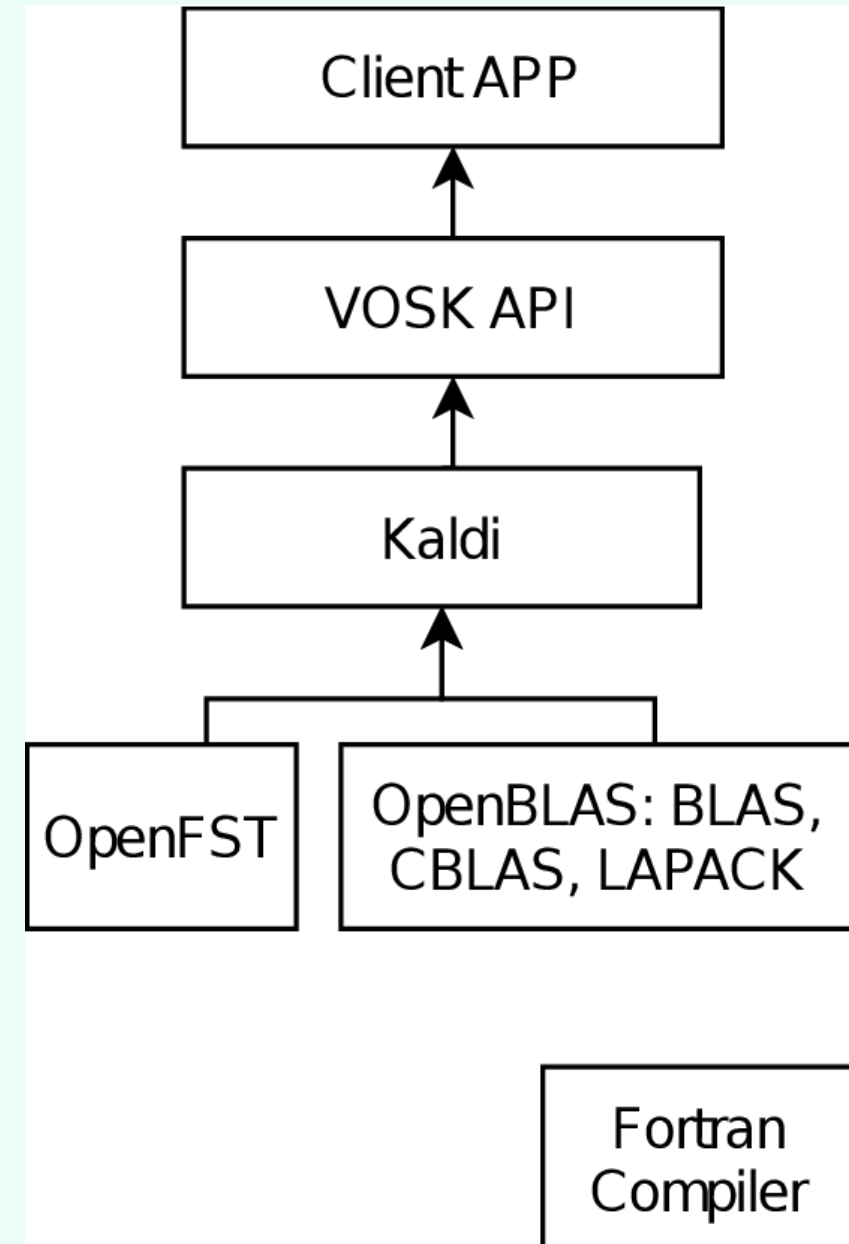
CMUSphinx	Kaldi	DeepSpeech	Facebook FlashLight	ESPNet
Лицензия BSD	Лицензия Apache	Mozilla Public Licence 2.0	Лицензия BSD	Лицензия Apache
Акустическая модель и словарь для русского языка	17 диалектов и языков, включая русский	Модель для русского языка не готова, подход требует больше данных	Модель для русского языка не найдена	Модель для русского языка не найдена
Статистическая модель языка	Модели языка в том числе и для мобильных устройств	Uses Google's TensorFlow project	Проект ориентирован на мощные вычислительные системы	Chainer and pytorch as a main deep learning engine
Проект был остановлен в пользу Kaldi в конце 2019	Внимание к расширяемости архитектуры проекта	Машинное обучение (публикация Baidu's Deep Speech)	Молодой исследовательский проект	Сквозной анализ речи

## ВОЗМОЖНОСТИ KALDI ASR

- › Распознавание речи на C++
- › Подобен Hidden Markov Model Toolkit (проприетарный аналог для работы с СММ)
- › Работа с конечными автоматами с выходом (FSTs) с помощью библиотеки OpenFst
- › Обширная работа с линейной алгеброй с помощью библиотек BLAS и LAPACK
- › Расширяемость (любой источник меры, например нейронная сеть)
- › Лицензия на СПО: Apache 2.0
- › Полноценные решение для распознавания речи
- › Публичный набор данных для русского языка от компании Alpha Cephei

## ОСОБЕННОСТИ СБОРКИ KALDI

- › Rolling-release, исследовательский проект
  - » Нет готовности поддерживать: releases, tags, versions
  - » Нет возможности поддерживать back-ports или какую-либо работу с неактуальными версиями
- › Относительно недавний переход с Make на Cmake
  - » Стоит уделить внимание на работу с зависимостями через внутренние средства загрузки пакетов
- › Зависимости от библиотек для линейной алгебры
  - » BLAS и LAPACK
  - » Компилятор Fortran



## КОМПИЛЯТОР FORTRAN

- › Имеет место не только для Kaldi ASR, но и для многих мат. проектов:
  - » Python NumPy
  - » Python SciPy
  - » PyTorch
- › Открытые (референсные) имплементации BLAS и LAPACK выполнены на Fortran
- › В ОС Аврора компилятор Fortran на данный момент отсутствует в силу оптимизаций сборки:
  - » Заметное снижение нагрузки на инфраструктуру сборки
  - » Недоступность сторонним разработчикам
  - » Приложение, собранное сторонним компилятором, получит ограничения на этапе проверки пакета
- › Проект F2C поддерживает стандарт F77, но не F95

## ВЫРАБОТАННЫЙ ПОДХОД

- › В текущей итерации воспользоваться сторонним компилятором Fortran (тем же GFortran из Linaro), или же исследовать порт на Си с инструментом f2c
- › Воспользоваться OpenBLAS (передав параметром MATHLIB), предоставить и OpenFST
- › Воспользоваться проектом VOSK:  
<https://alphacephei.com/vosk/index.ru>
  - » Минимальный рабочий пример
  - » Публичная модель русского языка в том числе и для мобильного ПО
- › Описанные наработки:  
<https://gitlab.com/AuroraOS/demos/asr-stt-project>

## ПОЛУЧЕННОЕ РЕШЕНИЕ

Кросс-компилятор Linaro GCC/GFortran 8.3 (той же версии, что и целевая система: например, 8.3 для Аврора SDK):

1. Разворачиваем rootfs (Аврора SDK)
  2. Укажем для OpenBLAS/LAPACK путь до sysroot и компиляторов
  3. Соберем OpenFST в той же среде с autoconf
  4. Kaldi соберем с CMake, указав OpenBLAS для параметра MATHLIB
  5. Воспользуемся примером из проекта VOSK на Си для теста
  6. В конечном приложении вызовем VOSK API в нативной среде на Qt/QML/Silica
- › Можно переиспользовать наработки для других POSIX-систем

<b>KaldiRecognizer</b>
+ sample_frequency
+ model
+ grammar
+ spk_model
+AcceptWaveform()
+PartialResult(...)
+Result()
+FinalResult()

<b>Model</b>
+ model_path
+ Ref(...)
+Unref()
+FindWord()

<b>SpkModel</b>
+ model_path
+ Ref(...)
+Unref()

## ПРОВЕРКА РЕЗУЛЬТАТОВ

### › INOI R7

- » Qualcomm Snapdragon 212
- » 4 ядра 1.2GHz
- » 2GB ОЗУ
- » ОС Аврора 3.2.2.20.

### › Без исследований по оптимизации:

- » Не мгновенно, но меньше 10 сек
- » ~ 100% первого ядра CPU
- » ~ 20% второго ядра







## **АЛЕКСЕЙ АНДРЕЕВ**

- › СТАРШИЙ ИНЖЕНЕР-РАЗРАБОТЧИК,  
ОТКРЫТАЯ МОБИЛЬНАЯ ПЛАТФОРМА
- › A.ANDREEV@OMP.RU

## **KIRILL CHUVILIN**

- › НАЧАЛЬНИК ОТДЕЛА РАЗВИТИЯ И ПОДДЕРЖКИ РАЗРАБОТЧИКОВ,  
ОТКРЫТАЯ МОБИЛЬНАЯ ПЛАТФОРМА
- › К.Т.Н., ДОЦЕНТ МФТИ
- › K.CHUVILIN@OMP.RU

