

Решения с открытым кодом для локальных голосовых помощников

Игорь Агамирзян

Октябрь 2025

Голосовые помощники: текущее состояние

- Disclaimer – это не научный доклад, а обзор доступных инженерных решений, проверенных на практике.
- Массовые системы: Siri, Alexa, Google Assistant, Алиса и др.
- Работают через облачные сервисы — требуется постоянное интернет-соединение.
- Недостатки: недоступность без сети, риски конфиденциальности, зависимость от поставщика.

Почему локальные решения становятся ВОЗМОЖНЫМИ

- Рост производительности одноплатных компьютеров (Raspberry Pi и др.).
- Оптимизация моделей машинного обучения.
- Открытые модели STT и TTS — можно запускать локально.
- Open-source экосистема ML позволяет обеспечить полную автономность.
- Tiny ML – можно ли так называть такие решения?

Компоненты локального голосового помощника

- STT (Speech-to-Text) — распознавание речи (Whisper от OpenAI).
- NLP (Natural Language Processing) — интерпретация смысла.
- TTS (Text-to-Speech) — синтез речи (Piper от OpenHome Foundation).
- Wake Word Detection — голосовая активация (OpenWakeWord).
- Интеграционный слой — протокол Wyoming от OpenHome Foundation объединяет все модули.

Whisper от OpenAI (STT)

- Открытая модель распознавания речи, поддержка десятков языков.
- Разные размеры моделей (tiny–small–large–turbo), возможность запуска на CPU.
 - К сожалению, драйверы Vulkan не поддерживают VideoCore VII GPU.
- Работает на Raspberry Pi 4/5, как и на любой другой Linux-системе.
- Существует стандартная реализация на Python и оптимизированная Whisper.cpp на C++.
 - В режиме CPU на Raspberry Pi 5 распознавание работает в несколько раз быстрее, чем на Intel Xeon 3.8 GHz.
- Поддержка протокола Wyoming через адаптер Whisper Satellite.

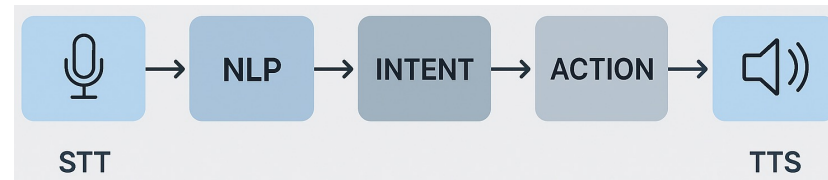
Piper от OpenHome (TTS)

- Синтезатор речи с открытым кодом, компактные модели ONNX.
- Работает офлайн, поддерживает множество голосов и языков.
 - Есть возможность обучения на свой голос.
- Интеграция с Home Assistant через Wyoming API.
- Может использоваться как локальный сервер синтеза речи.
 - Легковесный, работает очень быстро.
 - Wyoming API обеспечивает хэширование произнесённых фраз.

Протокол Wyoming — архитектура и особенности

- Разработан OpenHome Foundation для унификации использования голосовых компонентов.
- Использует TCP/IP сокеты, потоковую передачу аудио и текстовых сообщений.
- Каждый сервис (STT, TTS, WakeWord, NLP) — отдельный процесс/узел.
- Протокол определяет типы сообщений: CONNECT, EVENT, AUDIO, TEXT, RESPONSE.
- Home Assistant выступает как Wyoming Hub — маршрутизирует запросы.
- Обеспечивает независимость модулей и совместимость различных реализаций.

Система интентов (Intents) в голосовых помощниках



- NLP (Natural Language Processing) преобразует текст, полученный от STT, в структурированные команды.
- Intent — это шаблон, описывающий, что пользователь хочет сделать (например: включить свет, спросить погоду).
- Модель интентов включает:
 - Примеры фраз (training phrases);
 - Слоты/entities — переменные (например, комната, устройство, время).
- После распознавания интента генерируется JSON-запрос к системе автоматизации (например, Home Assistant).
- В Home Assistant используется движок `intent_script` и локальный `conversation agent`, обрабатывающий команды.
- Результат (ответ или подтверждение действия) передаётся обратно через TTS-модуль пользователю.

Примеры описания интенгов

```
language: "ru"
intents:
  HassTurnOn:
    data:
      - sentences:
          - "Зажги (<area> <name>|<name> <area>)"
          - "Включи (<area> <name>|<name> <area>)"
    requires_context:
      domain: "light"
    slots:
      domain: "light"
    response: "lights_area"

  HassTurnOff:
    data:
      - sentences:
          - "Погаси (<area> <name>|<name> <area>)"
          - "Выключи (<area> <name>|<name> <area>)"
    requires_context:
      domain: "light"
    slots:
      domain: "light"
    response: "lights_area"
```

```
language: "ru"
intents:
  HassTurnOn:
    data:
      - sentences:
          - "<turn_on> [весь] свет [здесь|в (комнате|помещении)]"
          - "<turn_on> [здесь|в (комнате|помещении)] [весь] свет"
    response: "lights_area"
    slots:
      domain: "light"
    requires_context:
      area:
        slot: true
```

```
language: "ru"
responses:
  intents:
    HassTurnOn:
      lights_area: "Включаю {{ slots.name }} {{ slots.area }}"
    HassTurnOff:
      lights_area: "{{ slots.name }} {{ slots.area }} выключен"
```

Пример конфигурации локальной голосовой системы

- **1** Raspberry Pi 4 (4 GB): Home Assistant + Wyoming Hub
 - Управляет интеграцией и маршрутизацией сообщений Wyoming.
 - Подключены Whisper (STT) и Piper (TTS) на Raspberry Pi 5 по протоколу Wyoming.
- **2** Raspberry Pi 5 (16 GB): Whisper + Piper + OpenWakeWord + Wyoming Satellite
 - Локальное распознавание и синтез речи, wake word активация.
 - Связь с Home Assistant по Wyoming через локальную сеть.
 - Одновременно служит голосовым помощником и обрабатывает STT/TTS-запросы
- **3** Raspberry Pi 4 (8 GB): OpenWakeWord + Wyoming Satellite
 - Передает аудио по Wyoming на Raspberry Pi 5 для STT/TTS обработки.
 - Служит голосовым помощником в другом помещении – реализует распределённую обработку и обеспечивает масштабируемость системы.
- Таким образом, определение активирующего слова происходит на каждом сателлите локально (аудио-поток передаётся на localhost), а передача аудио по локальной сети происходит только при STT/TTS-операциях.

Немного технических деталей

```
cd /home/igora/whisper.cpp
```

```
./build/bin/whisper-server \
```

```
--model models/ggml-small-q8_0.bin \
```

```
--audio-ctx 512 \
```

```
--no-timestamps \
```

```
--suppress-nst \
```

```
--language russian \
```

```
--no-gpu
```

```
cd /home/igora/wyoming-whisper-api-client
```

```
./script/run \
```

```
--uri tcp://0.0.0.0:7891 \
```

```
--api http://127.0.0.1:8080/inference
```

```
cd /home/igora/wyoming-openwakeword
```

```
./script/run \
```

```
--uri 'tcp://0.0.0.0:10400' \
```

```
--threshold 0.7 \
```

```
--custom-model-dir ./custom-model \
```

```
--preload-model glafira
```

```
cd /home/igora/wyoming-piper
```

```
./script/run \
```

```
--piper '/usr/share/piper/piper' \
```

```
--voice ru_RU-irina-medium \
```

```
--uri 'tcp://0.0.0.0:10200' \
```

```
--data-dir ./data \
```

```
--download-dir ./data \
```

```
--streaming
```

```
cd /home/igora/wyoming-satellite
```

```
./script/run \
```

```
--name 'Glafira Living Room' \
```

```
--uri 'tcp://0.0.0.0:10700' \
```

```
--mic-command 'arecord -r 16000 -c 1 -f S16_LE -t raw' \
```

```
--snd-command 'aplay -r 22050 -c 1 -f S16_LE -t raw' \
```

```
--wake-uri 'tcp://127.0.0.1:10400' \
```

```
--awake-wav './listen.wav' \
```

```
--mic-auto-gain 5 \
```

```
--mic-noise-suppression 2
```

Выводы и перспективы

- Open-source решения позволяют создать полностью автономных голосовых помощников.
- Wyoming обеспечивает гибкую модульную интеграцию STT, TTS, NLP, WakeWord.
- Рост вычислительных возможностей Raspberry Pi делает локальные решения реальностью.
- Будущее — развитие edge-архитектуры и расширение совместимости компонентов.