

# МИКРОСЕРВИСНЫЙ ПОДХОД К ПРОЕКТИРОВАНИЮ РЕЧЕВЫХ ИНТЕРФЕЙСОВ

**Житко В.А.**

*Белорусский государственный университет информатики и  
радиоэлектроники,*

*г. Минск, Республика Беларусь*

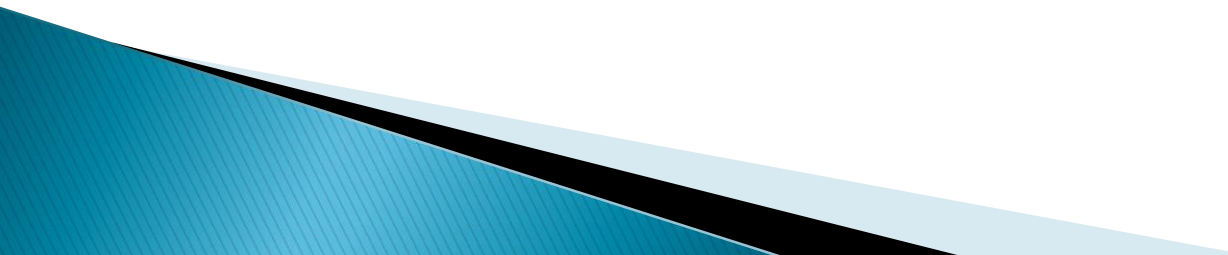
*[zhitko.vladimir@gmail.com](mailto:zhitko.vladimir@gmail.com)*

**Лобанов Б.М.**

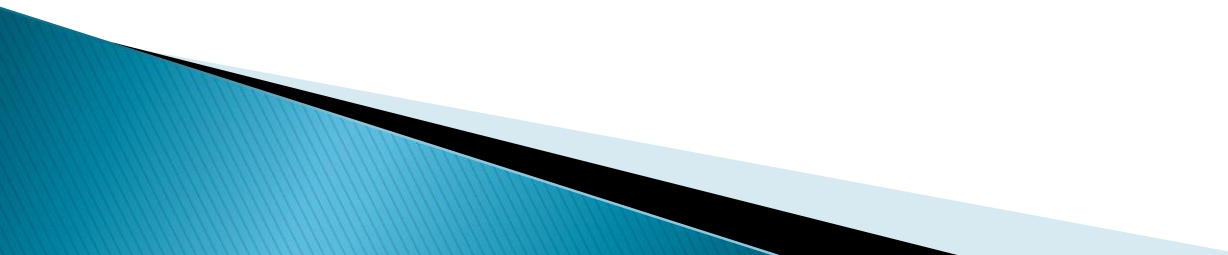
*Объединённый институт проблем информатики НАН Беларуси, Минск,  
Беларусь*

*[Lobanov@newman.bas-net.by](mailto:Lobanov@newman.bas-net.by)*

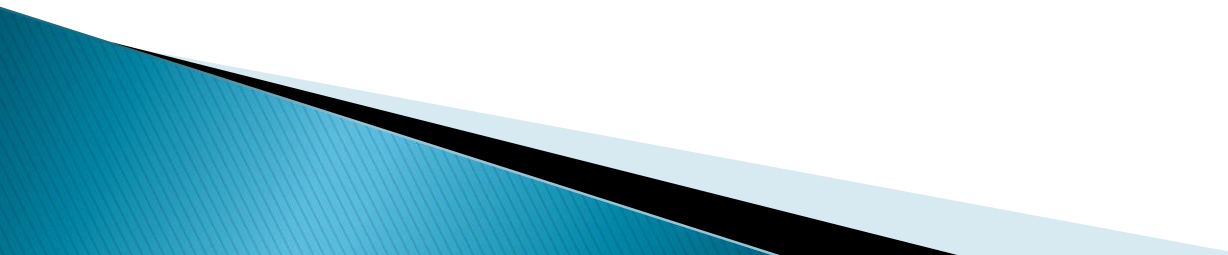
# Существующие программные продукты с речевым интерфейсом

- Apple Siri
  - Microsoft Cortana
  - Google Now
  - Amazon Alexa
  - Window–Eyes, NVDA, JAWS, Oralux
  - ...
- 

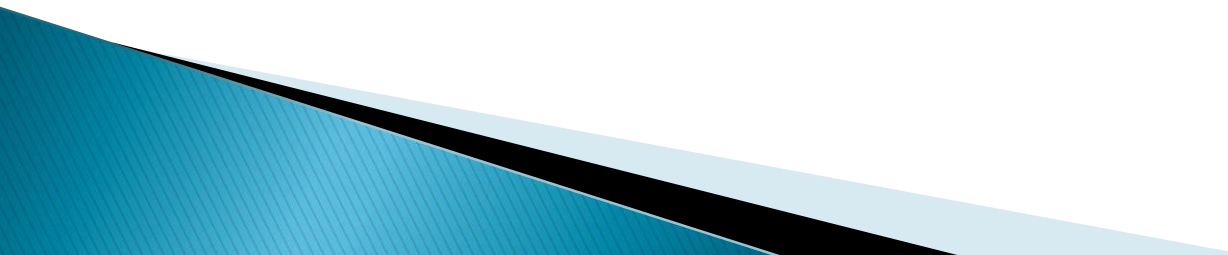
# Основные направления где используется речевой интерфейс

- Голосовые ассистенты (крупные корпорации Microsoft, Apple, Google)
  - Интеграция речевого ввода и вывода в колл-центре (call center)
  - Управление компьютером для слепых и плохо видящих
  - Озвучивание текстов (чтение книг, сообщений и пр.)
- 

# Структура приложений с речевым интерфейсом

- ↓ Распознавание речи
  - ↓ Анализ пользовательского сообщения
  - ↓ Выполнение действия, поиск решения и пр.
  - ↓ Синтез ответа
  - ↓ Синтез речи по тексту
- 

# Средства распознавания речи

- Google Web Speech API
  - Microsoft SAPI
  - Речевые технологии SpeechKit (Яндекс)
  - Voice Type Dictation, Voice Pilot и ViaVoice (IBM)
  - Dragon Dictate и Naturally Speaking (Dragon Systems)
  - Voice Assist (Creative Technology)
  - Listen for Windows (Verbex)
  - ...
- 

# Средства анализа текста

- АОТ (автоматическая обработка текста)  
(<http://www.aot.ru/>)
- Link Grammar Parser  
(<http://bobo.link.cs.cmu.edu/link/>)
- Russian Morphological Dictionary  
(<http://www.geocities.com/SiliconValley/Bit/1116/>)
- Технологии анализа и поиска текстовой информации (RCO) (<http://www.rco.ru/>)

# Средства анализа текста

- Russian Link Grammar Parser  
(<http://sz.ru/parser/>)
- Синтаксический анализатор естественного текста на русском языке (<http://syntax.ru/>)
- Langsoft (<http://langsoft.ch/>)
- Stanford Parser  
(<http://nlp.stanford.edu/software/lex-parser.shtml>)

# Средства интеллектуального анализа данных

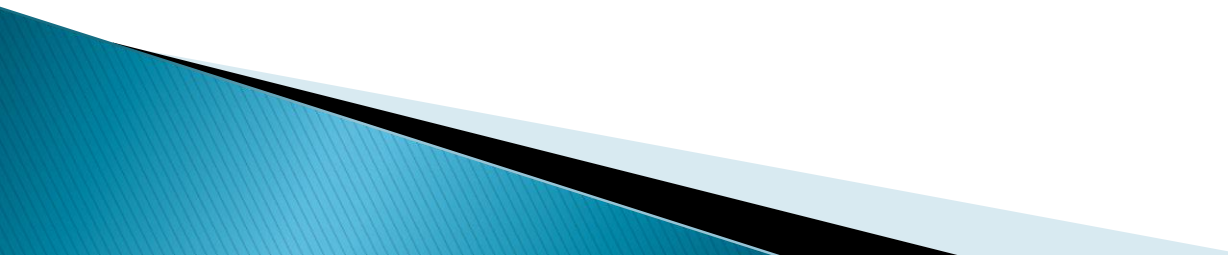
- ELKI (<http://elki.dbs.ifi.lmu.de/>)
  - Wolfram Language  
(<http://www.wolfram.com/language>)
  - KNIME (<http://www.knime.org/>)
  - Orange (<http://orange.biolab.si>)
  - Massive Online Analysis  
(<http://moa.cms.waikato.ac.nz/>)
- 



# Средства интеллектуального анализа данных

- Sentic API (<http://sentic.net/api/>)
- Weka  
(<http://www.cs.waikato.ac.nz/~ml/weka>)
- LIONsolver (<http://lionoso.com/>)
- OSTIS
- ...

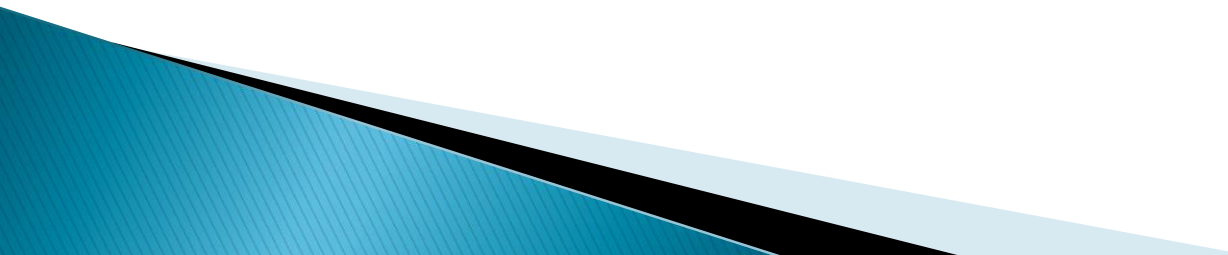
# Тезаурусы и базы знаний

- Русский ОбщеСемантический Словарь (РОСС)
  - Wiktionary (<http://wiktionary.org>)
  - RussNet (<http://project.phil.spbu.ru/>)
  - UNLWEB (<http://unlweb.net>)
  - WordNet (<http://wordnet.princeton.edu/>)
  - ...
- 

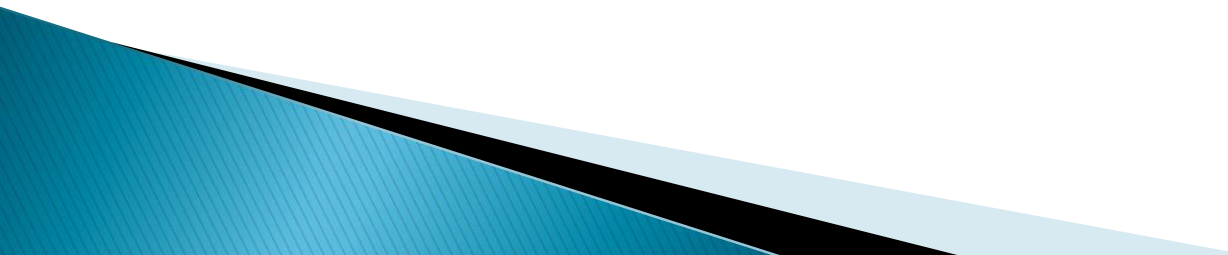
# Средства синтеза речи

- MultiPhone (<http://www.corpus.by/>)
- Microsoft Speech API  
(<https://msdn.microsoft.com/en-us/library/ms870025.aspx>)
- Festival Speech Synthesis System  
(<http://www.cstr.ed.ac.uk/projects/festival/>)
- FreeTTS (<http://freetts.sourceforge.net/>)
- eSpeak (<http://espeak.sourceforge.net/>)
- Vocaloid (<http://www.vocaloid.com>)

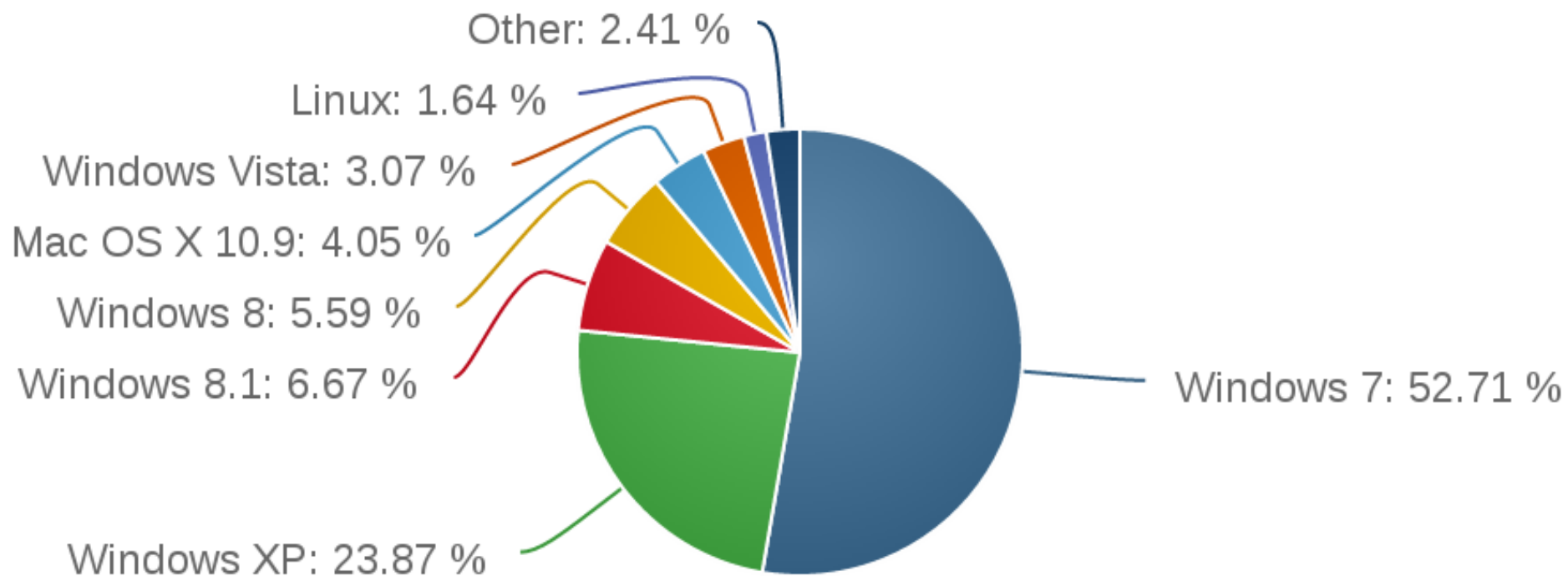
# Проблемы

- Отсутствие единого подхода в реализации составляющих компонентов речевого интерфейса
  - Отсутствие технологии интеграции разнородных компонентов
  - Отсутствие спецификации и перечня типовых компонентов
  - *Сложность в разработке и использовании различных сторонних разработок*
- 

# Подход к построению речевых приложений

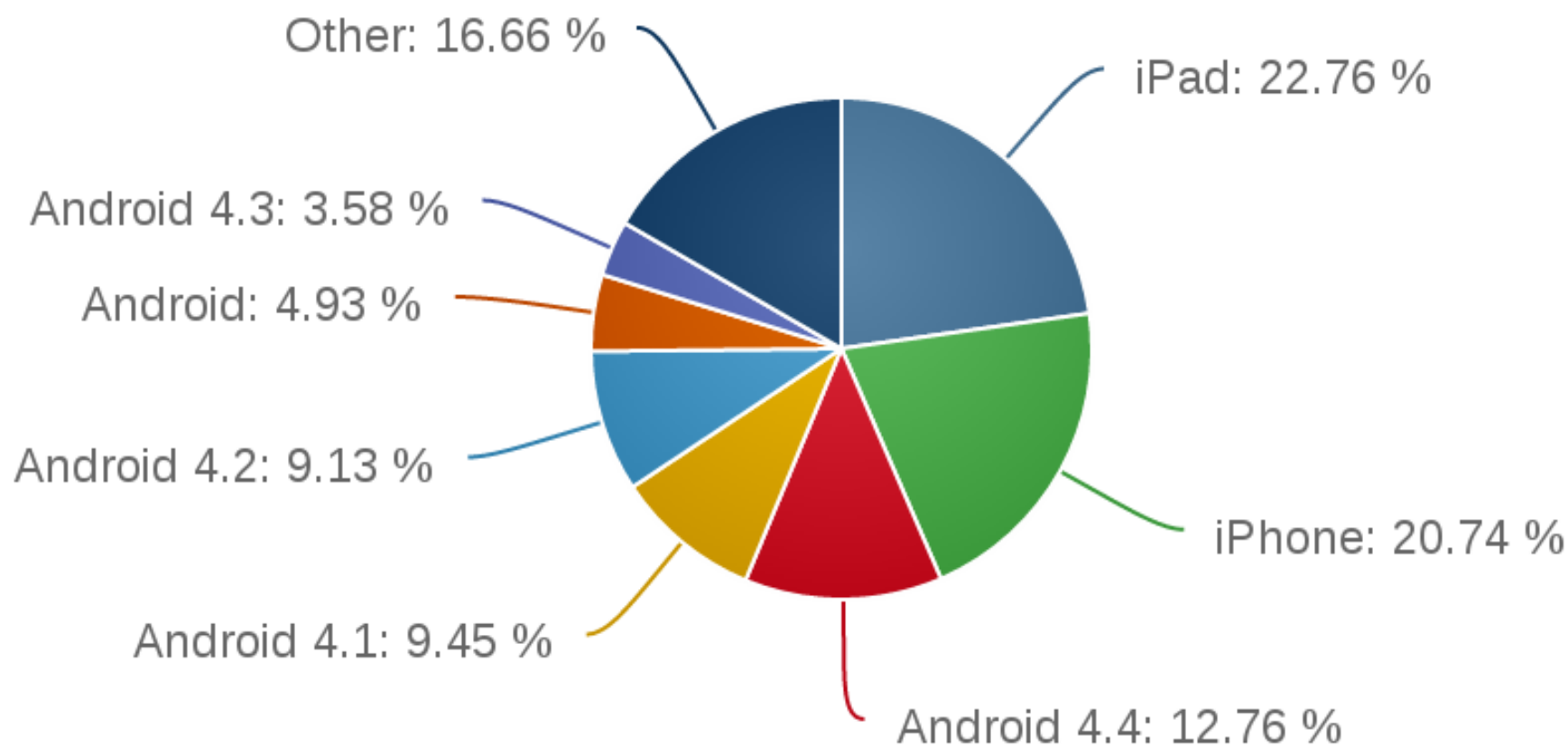
- Сервис-ориентированная архитектура
  - Использование микросервисов
  - Выделение бизнес логики в отдельные сервисы
  - Кроссплатформенность
- 

# Кроссплатформенность (настольные компьютеры)



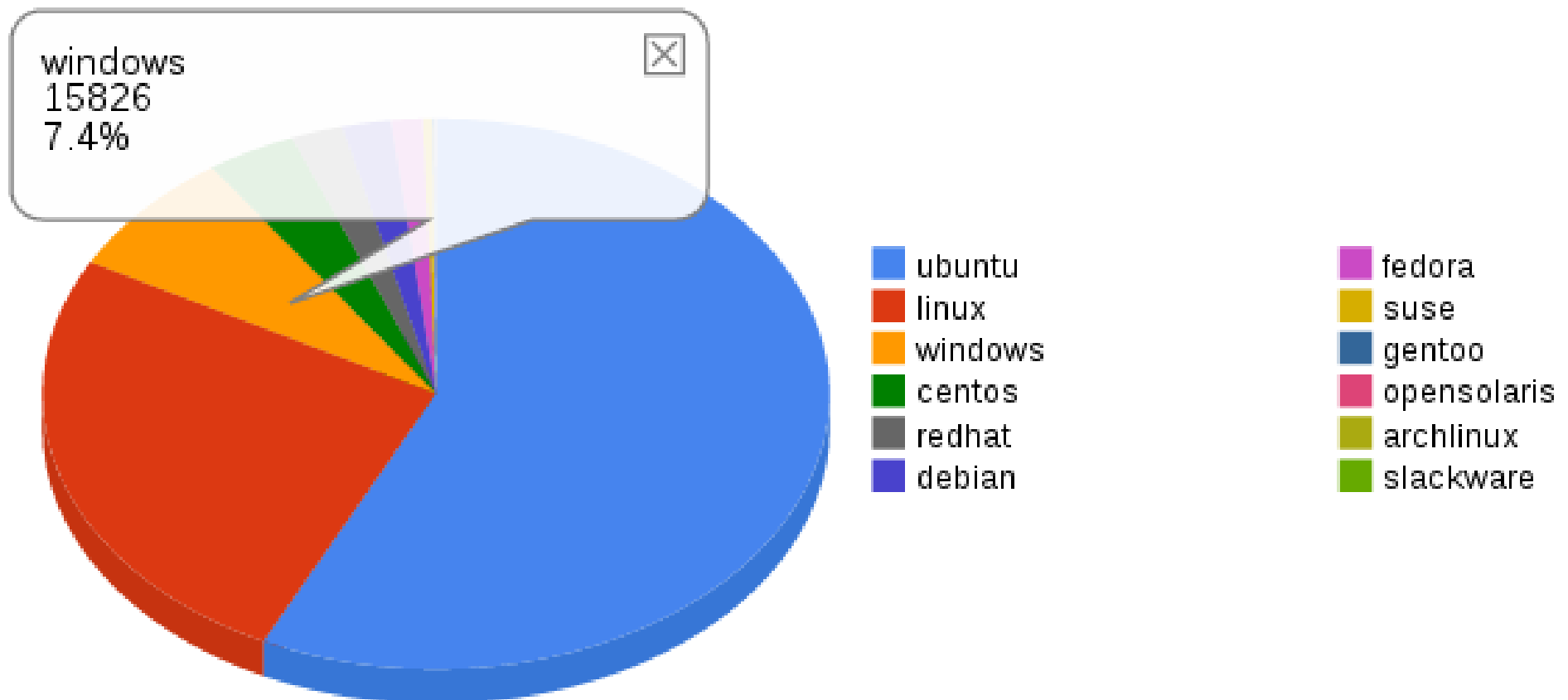
<http://www.netmarketshare.com/>

# Кроссплатформенность (мобильные устройства)



<http://www.netmarketshare.com/>

# Кроссплатформенность (сервера)





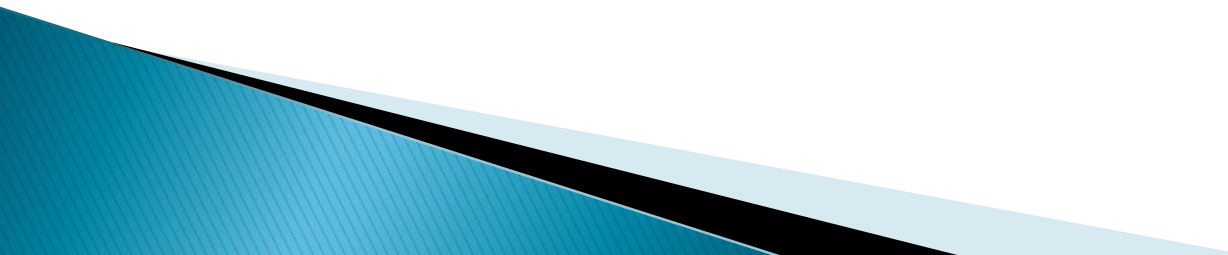
# Сервис-ориентированная архитектура

- ▶ Сервис-ориентированная архитектура (SOA, англ. service-oriented architecture) — *модульный подход к разработке программного обеспечения, основанный на использовании распределённых, слабо связанных (англ. loose coupling) заменяемых компонентов, оснащённых стандартизированными интерфейсами для взаимодействия по стандартизированным протоколам.*

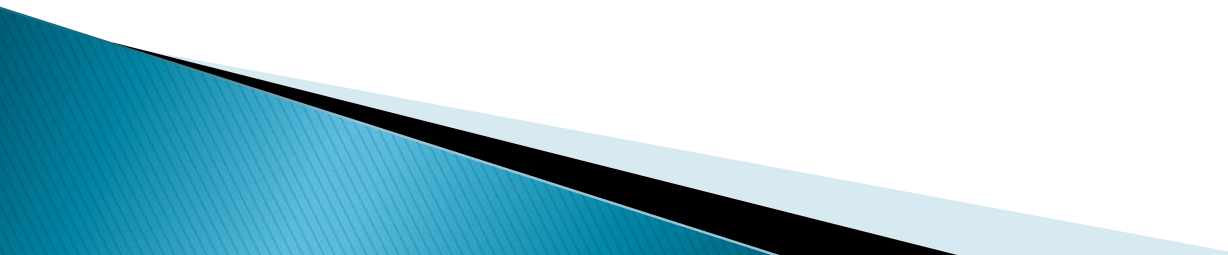
# Микросервисы

- ▶ Микросервисы — *это архитектурный стиль, при котором сложное приложение разбивается на мелкие, независимые процессы, взаимодействующие через методы API, независимых от языка.*
- сервисы определяются бизнес-задачами
- логика реализована в методах сервиса
- каждый микросервис решает задачи, связанные с единственным или несколькими неразделимыми объектами домена

# Преимущества подхода

- **масштабируемость** – микросервисную архитектуру очень легко масштабировать;
  - **отказоустойчивость** – надёжность достигается за счёт запуска дополнительных процессов;
  - **обновляемость** – малая связанность сервисов и их относительная простота делает процесс тестирования и обновления технологий довольно простой задачей.
  - **скорость проектирования и прототипирования**
- 

# Задачи

- Определить перечень и функционал сервисов (в рамках микросервисного стиля)
  - Определить протокол общения сервисов
  - Реализация базового набора сервисов
- 

# Экспериментальная система с речевым интерфейсом

The image displays the 'Speech Apps' experimental system interface. It features a main menu on the left with icons for 'Телеграф' (Telegraph), 'Сказка' (Fairy Tale), 'Перевод' (Translation), and 'Калькулятор' (Calculator). The main menu is titled 'Главное меню' (Main Menu).

Three application windows are shown:

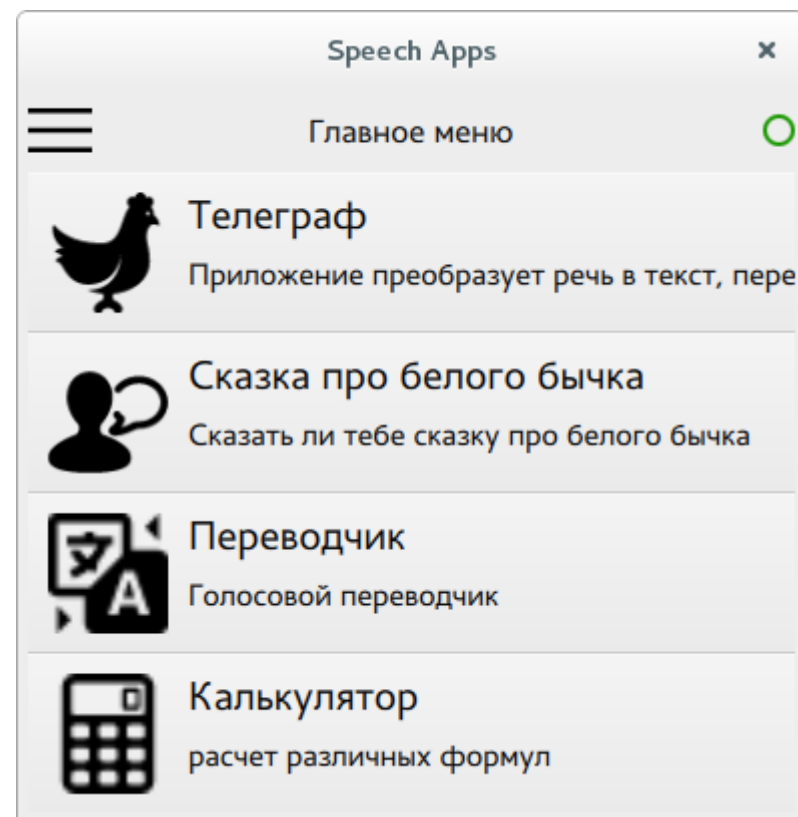
- Переводчик (Translator):** This window shows a conversation between a user and a computer. The user selects 'русский' (Russian) and 'American English'. The computer responds with 'Hello! let's start work!'. The user says 'Привет' (Hello), and the computer translates it to 'Hi'. The user asks 'Как дела' (How are you), and the computer translates it to 'How are you'. The user asks 'Какая сейчас погода' (What is the weather now), and the computer translates it to 'What is the weather'. The user says 'сейчас солнечно' (it's sunny now).
- Калькулятор (Calculator):** This window shows a conversation between a user and a computer. The user says 'Могу рассчитать любую формулу' (I can calculate any formula). The computer responds with 'Могу рассчитать любую формулу' (I can calculate any formula). The user asks 'найти сумму 2 и 5 все умножить на 10' (find the sum of 2 and 5, then multiply by 10). The computer calculates  $(2+5)*10 = 70$ . The user asks '2 + 2'. The computer calculates  $2+2 = 4$ . The user asks '5 умножить на сумму 2 и 3 и поделить на 2' (5 multiplied by the sum of 2 and 3, then divided by 2). The computer calculates  $(5*(2+3))/2 = 12.5$ .
- Сказка (Fairy Tale):** This window shows a conversation between a user and a computer. The user says 'Сказка о белом бычке' (Fairy tale about a white bull). The computer responds with 'Хочешь? Я Расскажу тебе о белом бычке.' (Do you want? I will tell you about a white bull). The user says 'Да расскажи' (Yes, tell me). The computer responds with 'Ты Да расскажи мне о белом бычке.' (You Yes, tell me about a white bull). The user says 'Таня' (Tanya).

The 'Speech Apps' window also shows settings for 'Устройство аудио входа' (Audio input device) and 'Устройство аудио выхода' (Audio output device), both set to 'H600 [Wireless Headset]'. It also shows settings for 'Языки' (Languages) and 'Интерфейс' (Interface), both set to 'русский' (Russian). It also shows settings for 'Синтез речи' (Speech synthesis) and 'Распознавание речи' (Speech recognition), both set to 'Russian' and 'русский' respectively.

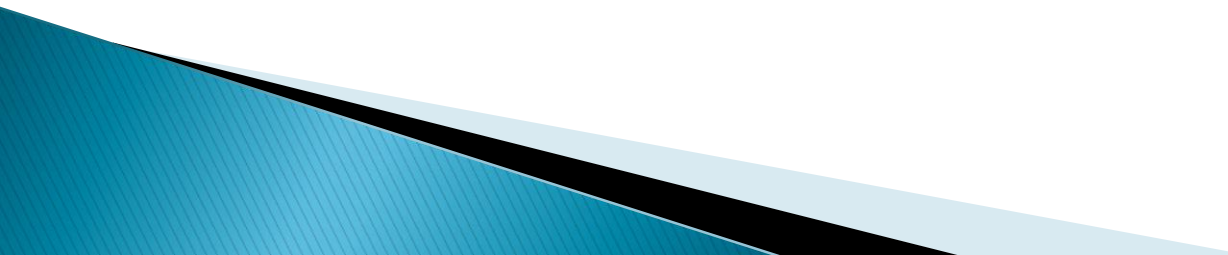
# Управление приложением ГОЛОСОМ

## Возможности:

- запуск модулей
- ВЫХОД в основное меню
- завершение работы приложения



# Пример диалога управления приложением голосом

- ▶ *Пользователь: начнём пожалуй*
  - ▶ Система: выберете приложение
  - ▶ *Пользователь: запустить переводчик*
  - ▶ Система: запуск переводчика
  - ▶ ...
  - ▶ *Пользователь: команда назад*
  - ▶ *Пользователь: завершить работу*
  - ▶ Система: ещё увидимся
- 

# Телеграф

- ▶ Распознает речевое сообщение пользователя и дублирует его синтезатором речи
- ▶ Возможное использование:
- ▶ *Передача речевых команд по узкополосному каналу информации в форме закодированного телеграфного сообщения с воспроизведением речи синтезатором голоса на приёмном конце*



# Пример диалога

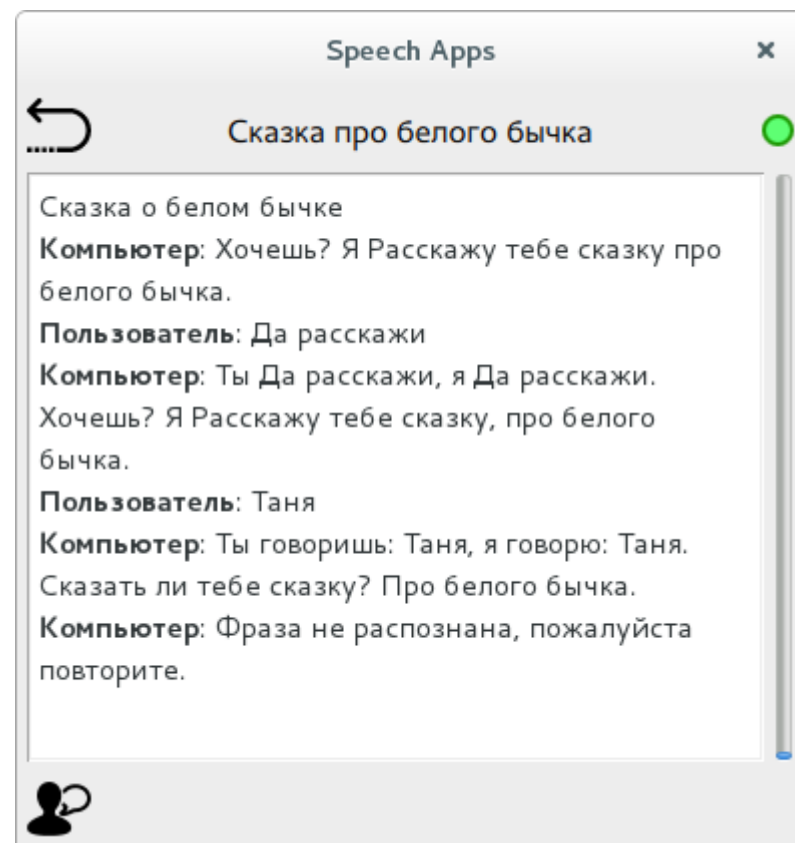
- ▶ *Пользователь: Прицел 6-30 – четыре мины, прицел 6-37 – две мины, прицел 6-22 – две мины*
- ▶ *Пользователь: Прицел 6-30 смещение правее на 2*

# Бычок (сказка)

Демонстрация простейшего диалога, где в тексте ответа используются элементы запроса

Возможное использование:

► *Приложение телефонного сервиса развлечения для детей*



# Пример диалога

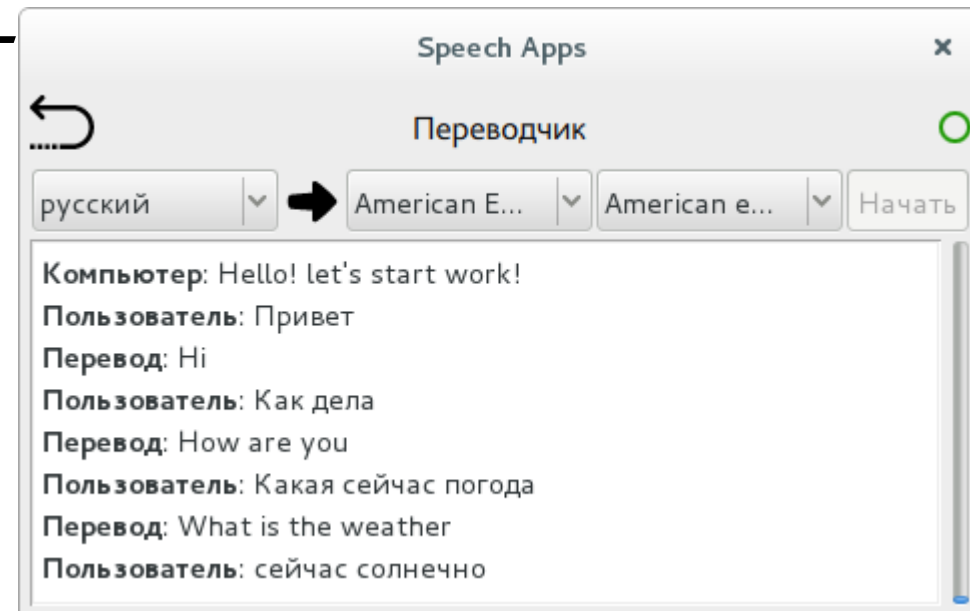
- ▶ Система: Хочешь? Я Расскажу тебе сказку про белого бычка.
- ▶ *Пользователь: да*
- ▶ Система: Ты *да*, я *да*. Хочешь? Я Расскажу тебе сказку, про белого бычка.
- ▶ *Пользователь: нет не надо*
- ▶ Система: Ты говоришь: *нет не надо*, а я говорю: *нет не надо*. Давай, я тебе сказку про белого бычка расскажу!

# Переводчик

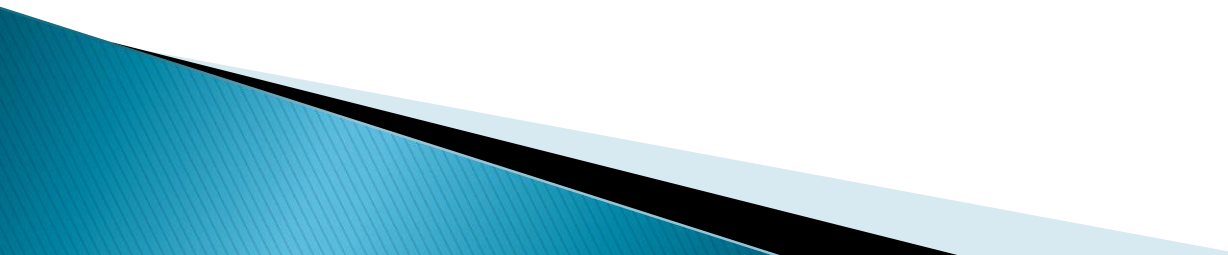
Перевод речевого сообщения пользователя на другой язык и воспроизведение результата используя речевой синтезатор.

Возможное использование:

- ▶ *Приложение ассистент при изучении иностранного языка*



# Пример диалога

- ▶ *Пользователь: добрый день*
  - ▶ Система: добры дзень
  - ▶ *Пользователь: сегодня солнечно*
  - ▶ Система: сёння сонечна
  - ▶ ...
  - ▶ *Пользователь: goodbye*
  - ▶ Система: да пабачэння
- 

# Калькулятор

- ▶ Демонстрация простейшего диалога с элементами поиска и расчёта ответа



# Пример диалога

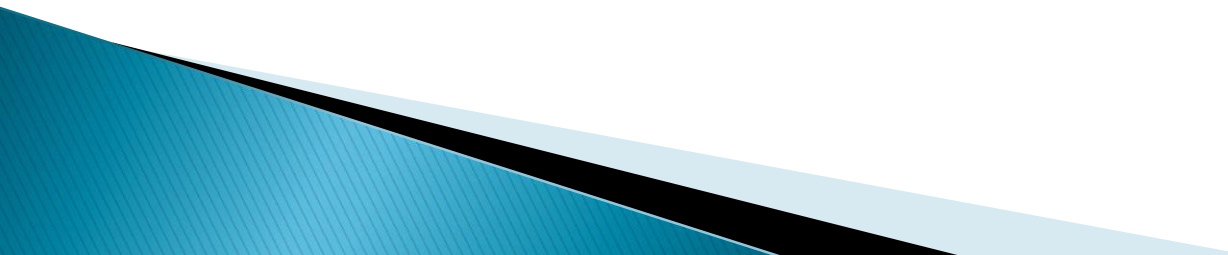
- ▶ *Пользователь: сложить 2 и 3*
- ▶ Система: Результат вычисления, равен 5
- ▶ *Пользователь: найти произведение 3 и 5*
- ▶ Система: Результат вычисления, равен 15
- ▶ *Пользователь: найти сумму 5 и 2 затем умножить результат на 10 и добавить 3*
- ▶ Система: Результат вычисления, равен 73

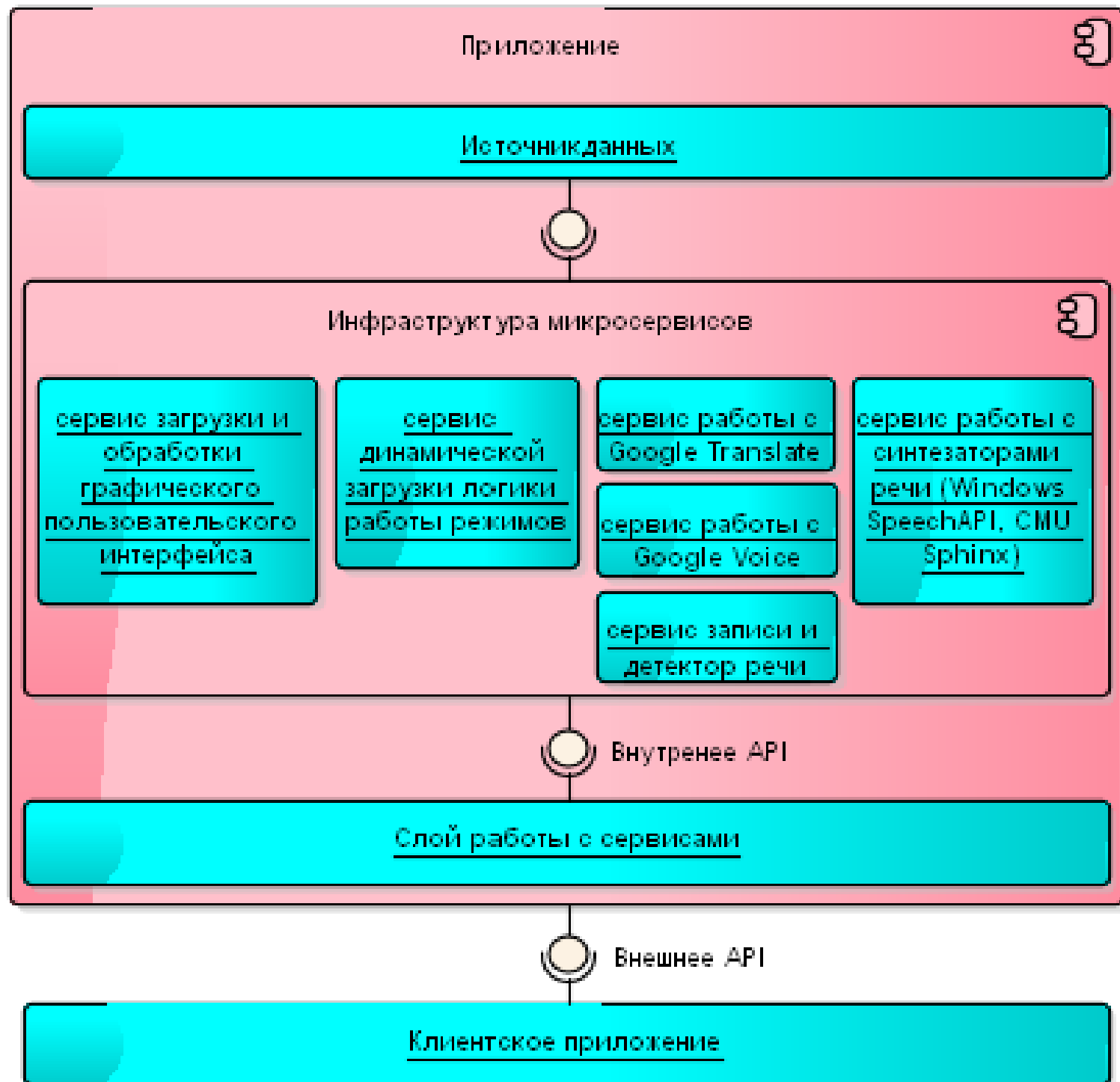
# Технологии

- Qt Framework (<http://www.qt.io/>)
  - C++, qml, javascript
- Speech Signal Processing Toolkit (SPTK) (<http://sp-tk.sourceforge.net/>)
- OpenAL (<http://sourceforge.net/projects/openal-soft/>)
- ...

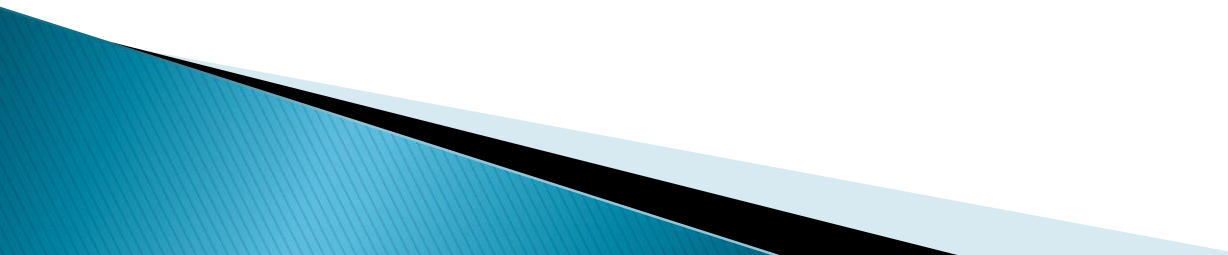


# Логическая структура приложения

- ↓ слой доступа к данным
  - ↓ слой поддержки сервис-ориентированной архитектуры (инфраструктура микросервисов)
  - ↓ слой служебных сервисов
  - ↓ слой клиентских приложений (в рамках реализуемого приложения это режимы)
- 



# Сервисы в экспериментальной системе

- сервис записи и детектор речи;
  - сервис работы с Google Voice;
  - сервис работы с Google Translate;
  - сервис работы с синтезаторами речи (Windows SpeechAPI, CMU Sphinx)
  - сервис загрузки и обработки графического пользовательского интерфейса
  - сервис динамической загрузки логики работы режимов
- 

# Пример реализации подключаемого модуля

```
import FileIO 1.0
import "Utils"
import "external/mathjs/math.js" as Mathjs
import "external/urim/Snowball.js" as Snowball
import "external/PEGjs/peg-0.8.0.js" as PEG
Item {
  property string id: "calculatorLogic"
  FileIO {
    id: grammar
    source: "./logic/calculator/grammar.peg"
    onError: console.log(msg)
  }
}
```

# Пример реализации подключаемого модуля

```
function parse(text) {  
    var stemmedWords = stemmerThem(inWords)  
    var processedWords = []  
    for (var i=0; i<stemmedWords.length; ++i) {  
        var word = stemmedWords[i].toLowerCase()  
        var operator = operatorMap[word]  
        if (!!operator) processedWords.push(operator)  
        else if (!isNaN(word))  
            processedWords.push(word)  
    }  
    if (!parser) parser =  
        PEG.PEG.buildParser(grammar.read())  
}
```

# МИКРОСЕРВИСНЫЙ ПОДХОД К ПРОЕКТИРОВАНИЮ РЕЧЕВЫХ ИНТЕРФЕЙСОВ

Спасибо за внимание

Исходный код:

- ▶ <https://github.com/zhitko/speech-apps>
  - ▶ <https://github.com/zhitko/sptk-analyzer>
  - ▶ [https://github.com/zhitko/QAS\\_RR](https://github.com/zhitko/QAS_RR)
- 