

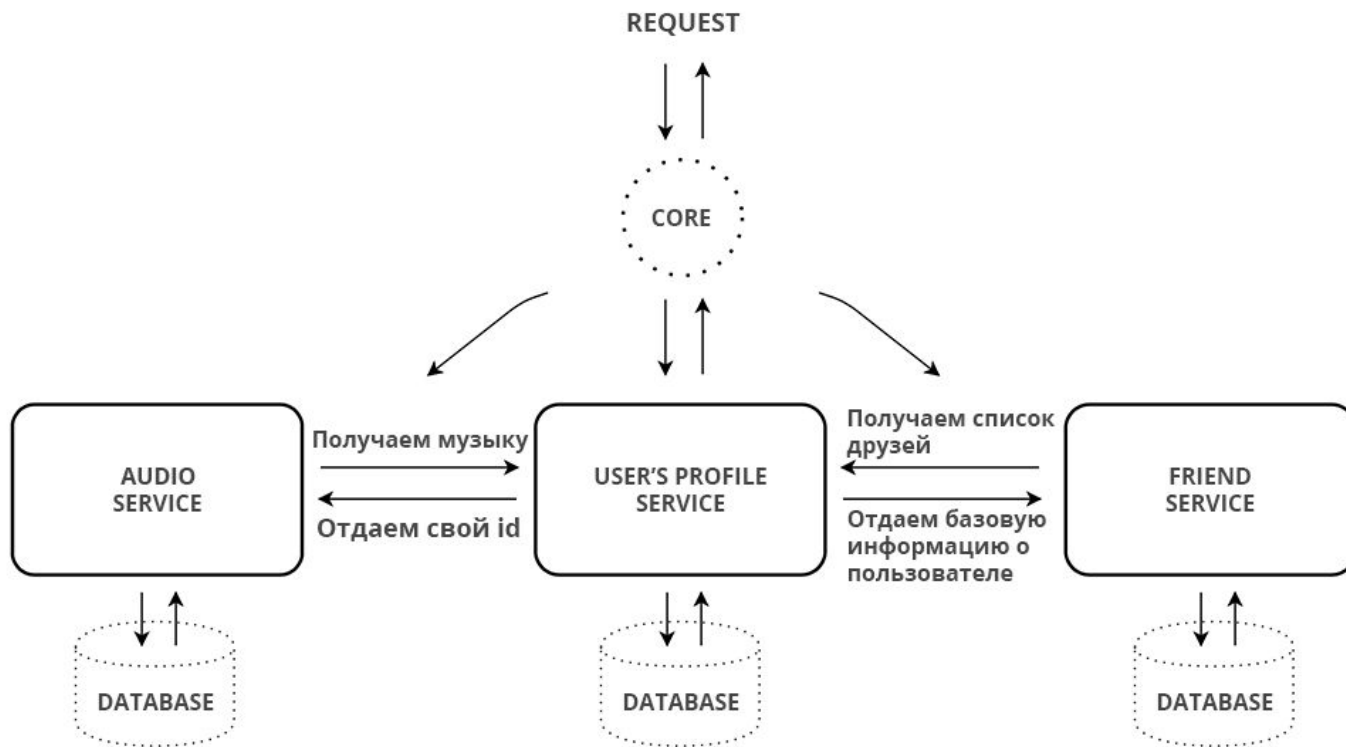
Профили пользователей

# Микросервис “Профили пользователей”

Задачи:

- Отображение информации профиля конкретного пользователя
- Передача базовой информации другим микросервисам (статус, ФИО и т.д.)
- Создание, удаление пользователя
- Изменение информации профиля
- Поиск пользователей с фильтрацией
- Связующее звено между микросервисами

# Архитектура взаимодействия



# Данные, относящиеся только к профилю

Education
id
graduationYear
country
city
university
department
modeOfStudy
status

PersonalViews
id
politicalView
worldView
smokingView
alcoholView
perPriority
implnOther
inspiredBy

BasicInfo
id
avatarId
firstName
secondName
sex
relStatus
birthday
hometown

Interests
id
activities
interests
favMusic
favBooks
favGames
favQuotes
about

ContactInfo
id
country
city
mobile
website
instagramAcc
skypeAcc
twitterAcc

# Хранение данных

Самые популярные юзеры:

FIRST LAYER: RAM

CACHE: LFU

CAPACITY: N

Менее популярные юзеры:

SECOND LAYER: DRIVE

CACHE: LFU

CAPACITY: M>N

“Забытые” юзеры:

THIRD LAYER: DATABASE

CAPACITY: K>M

# Объемы

- Количество пользователей - величина непредсказуемая
- Предельная верхняя оценка - 100 млн .. 500 млн .. 1 млрд
- На каждого пользователя от 5 (минимальная функциональность) до 40 (расширенная) полей (несколько таблиц)

# Ограничения

- Информация о конкретном пользователе должна возвращаться без задержек
- Используем load balancer
- Примерное ограничение - 15мс

# Масштабируемость

- Количество пользователей растет => растет количество запросов
  - Растет количество запросов => растет нагрузка
  - Растет нагрузка => масштабируем
- 
- Используем горизонтальный шардинг
  - Разбиваем по user\_id



# Отказоустойчивость

- Отказ шарда => проблема
- Повышаем надежность => репликация
- Для каждого шарда выделяем свой слейв

# API

Profile[] getProfiles(int[] ids)

int[] ids - список id профилей	return: Профили по заданным id
--------------------------------	--------------------------------

Profile getProfile(int id)

int id - id пользователя	return: Профиль по заданному id
--------------------------	---------------------------------

void setStatusOnline(int id)

int id - id пользователя	Обновляет время активности
--------------------------	----------------------------

# API

int getUserStatus(int id)

int id - id пользователя	return: Время последней активности
--------------------------	------------------------------------

Profile createProfile(String firstName, String lastName)

String firstName, String lastName - имя и фамилия	return: Созданный профиль
--	---------------------------

boolean createProfile(Profile profile)

Profile profile - профиль	return: Передан новый пользователь
---------------------------	------------------------------------

# API

`void removeProfile(int id)`

<code>int id</code> - id пользователя	Удаляет пользователя по id
---------------------------------------	----------------------------

`void removeProfile(Profile profile)`

<code>Profile profile</code> - профиль	Удаляет пользователя по профилю
--	---------------------------------

`void updateProfile(Profile profile)`

<code>Profile profile</code> - профиль	Обновляет указанный профиль или создает
--	---