

# Realm

кросс-платформенная  
мобильная база данных

Ткаченко Никита  
БПИ197

## Задачи:

- познакомиться с фреймворком
- определить основной функционал
- сравнить с альтернативами
- рассмотреть способы интеграции в проект

## Итог:

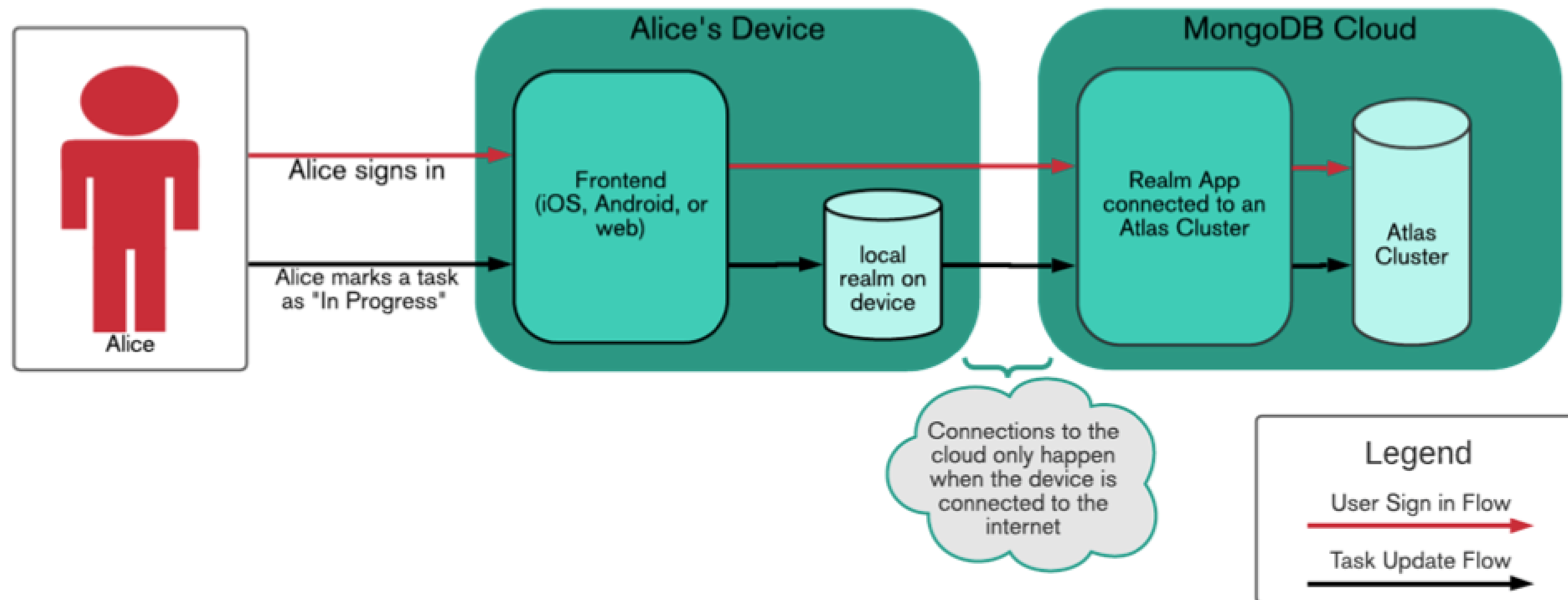
- демонстрация демо-приложения с Realm
- формирование основных выводов о технологии

# База данных Realm

- 1 Локальное хранилище
- 2 No-SQL
- 3 ООП модель без ORM
- 4 Live objects
- 5 Надежность сети



# Архитектура Realm



# Основной функционал

- 1 Создание объекта модели Realm (функц-ют как обычный объект в Swift)
- 2 Автоматическое обновление представление при изменении свойств объекта Realm
- 3 Запись объекта в базу (синхронно, блокирует поток)
- 4 Чтение объекта из базы (не блокирует )
- 5 Кодирование ключ-значение (изменять значение по ключу не бегая по коллекции)



# Основной функционал

- 6 Удаление объекта из базы
- 7 Различные запросы к базе для фильтраций и сортировок
- 8 Подпись на уведомление об изменении

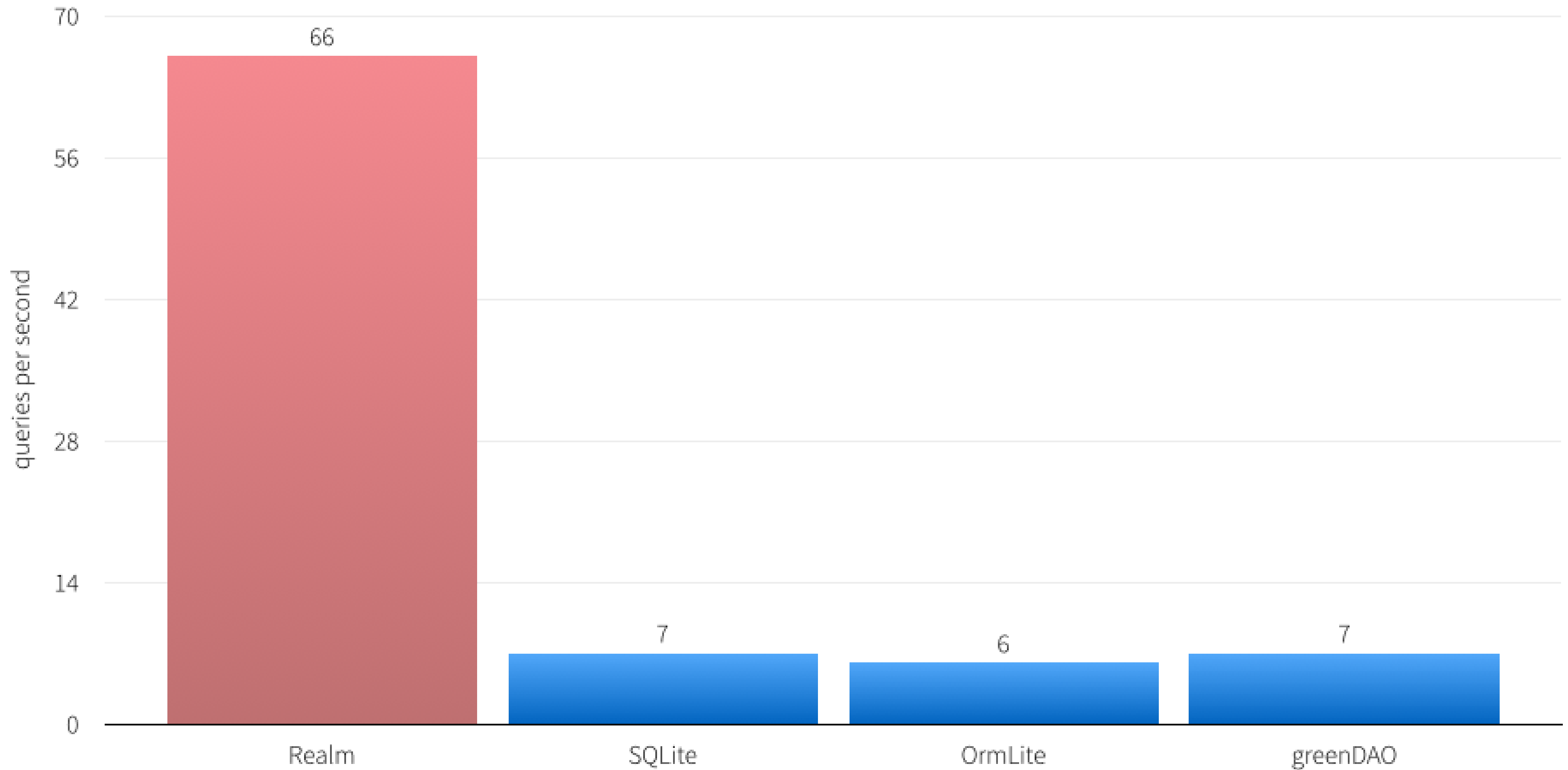


# Преимущества Realm

- 1 Кросс-платформенность
- 2 Скорость
- 3 Масштабируемость
- 4 Надежность
- 5 Простота

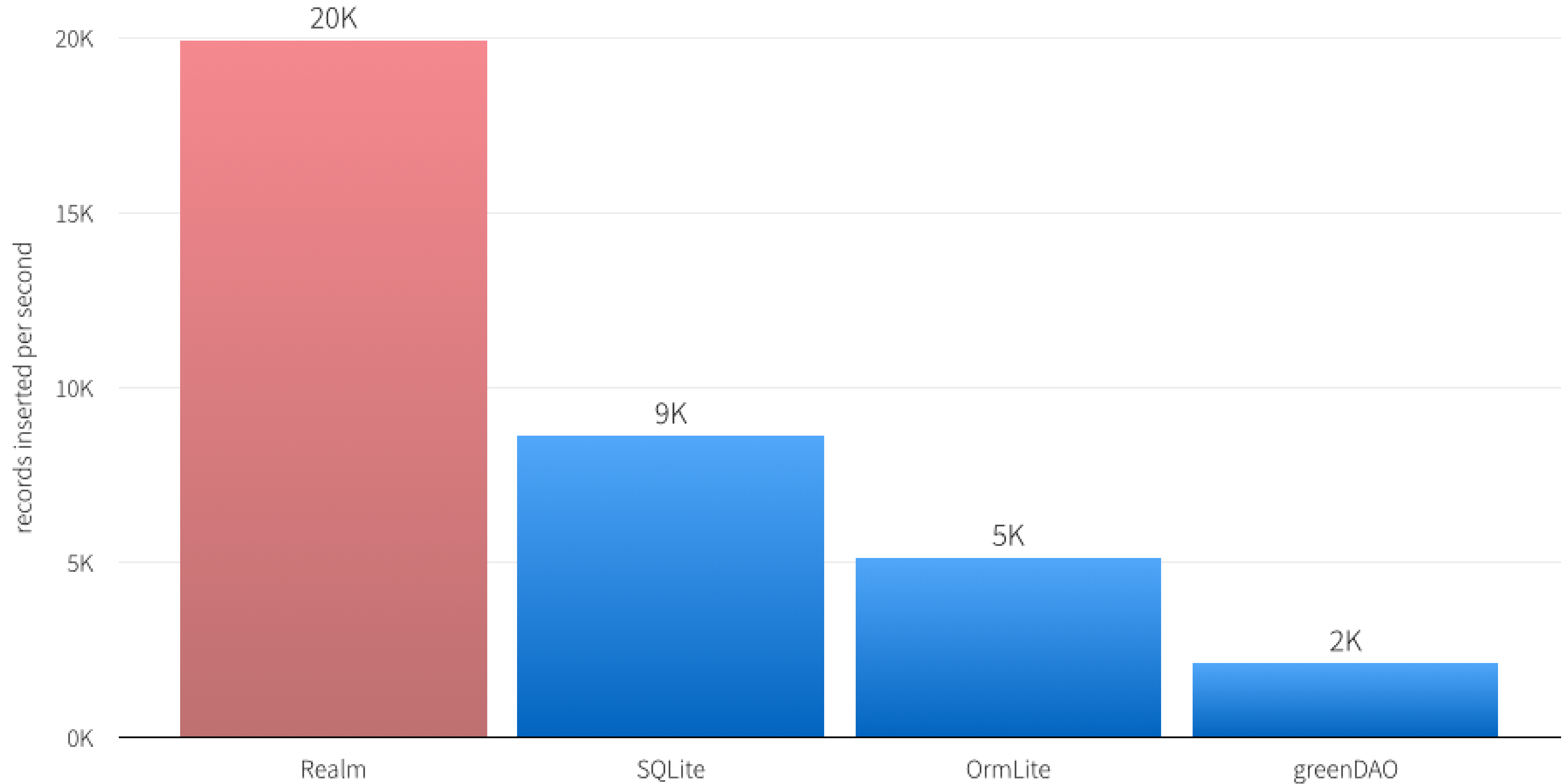
# Counts

*Get count of records matching a query on a database of 100k records (higher is better)*



# Inserts

*Insert 100k records, in a single transaction (higher is better)*





# Способы интеграции в проект

## 3 CocoaPods

Установить CocoaPods 1.10.0 или более поздней версии.

---

Запустите `pod repo update`

---

В подфайл добавить `use_frameworks!` и подключить "RealmSwift"

---

В терминале запустить `pod install`

---

## 1 Swift PM

## 2 Carthage

## 3 CocoaPods

# Демонстрация

1

Демонстрация приложения

2

Демонстрация Realm Browser

# Список источников

- Подключение Realm Sync - <https://docs.mongodb.com/realm/tutorial/ios-swift/>
- Документация по Realm - <https://docs.mongodb.com/realm-legacy/docs/swift/latest/index.html#prerequisites>
- Статья про Realm - <https://habr.com/ru/post/328418/>