



Данные

Бернгардт Григорий, руководитель iOS направления

Данные в приложениях

Работа с данными

Хранение данных

Данные в приложениях

Работа с данными

Хранение данных

Данные

Все, что наполняет наше
приложение

MTS RUS LTE18:33

Старый Город

Great

\$\$

Pastries

Delivery

European

Wine

Shisha

Live

ул. Пушкинская, 2, Воронеж, 394036

0,2 km

Пн-Вс 12.00-24.00

<http://cafegorod.ru>

Call
+7 (473) 255-39-82

Plan route

Photo of the location

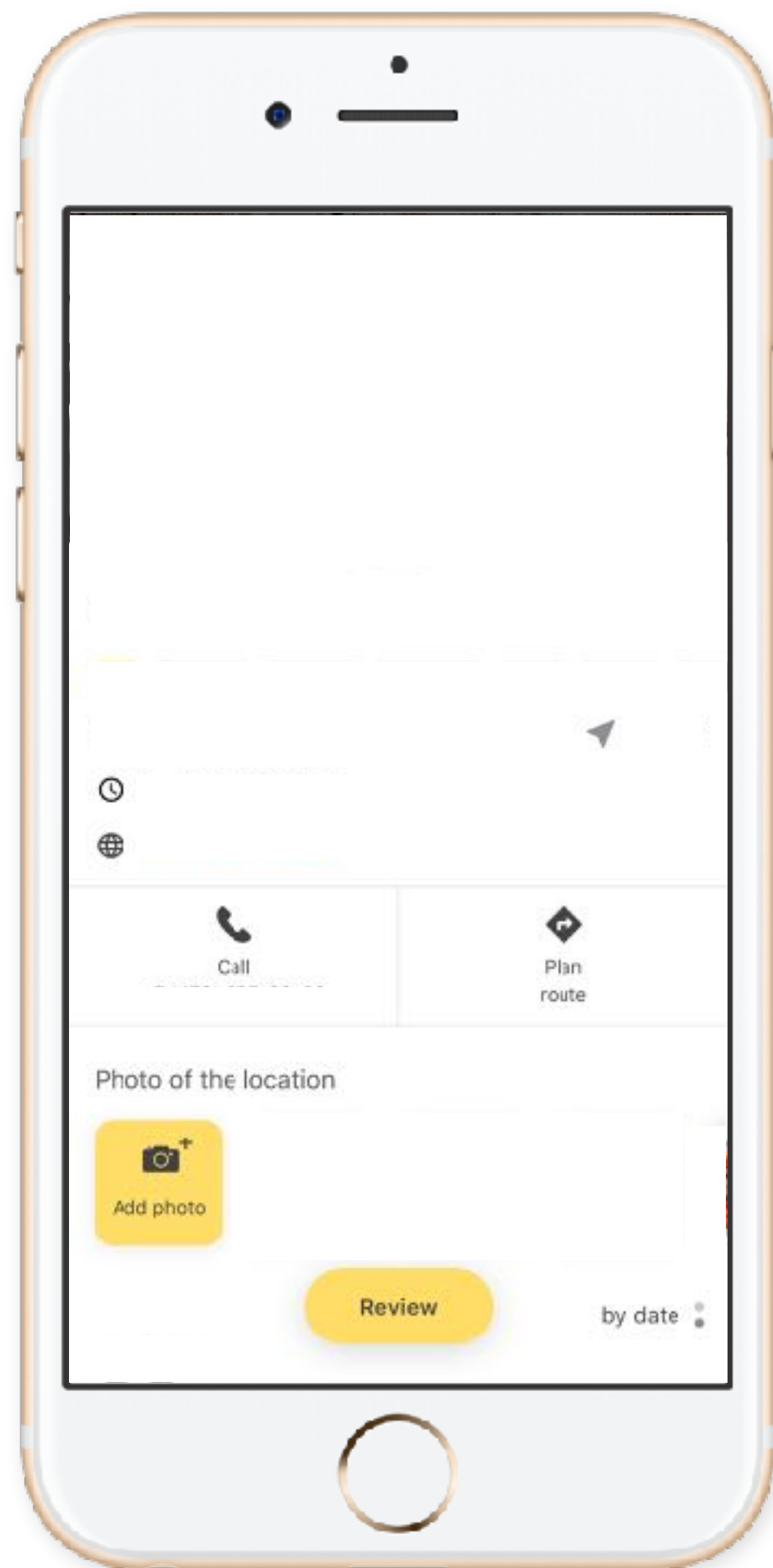
5 photos

Add photo

4 reviews

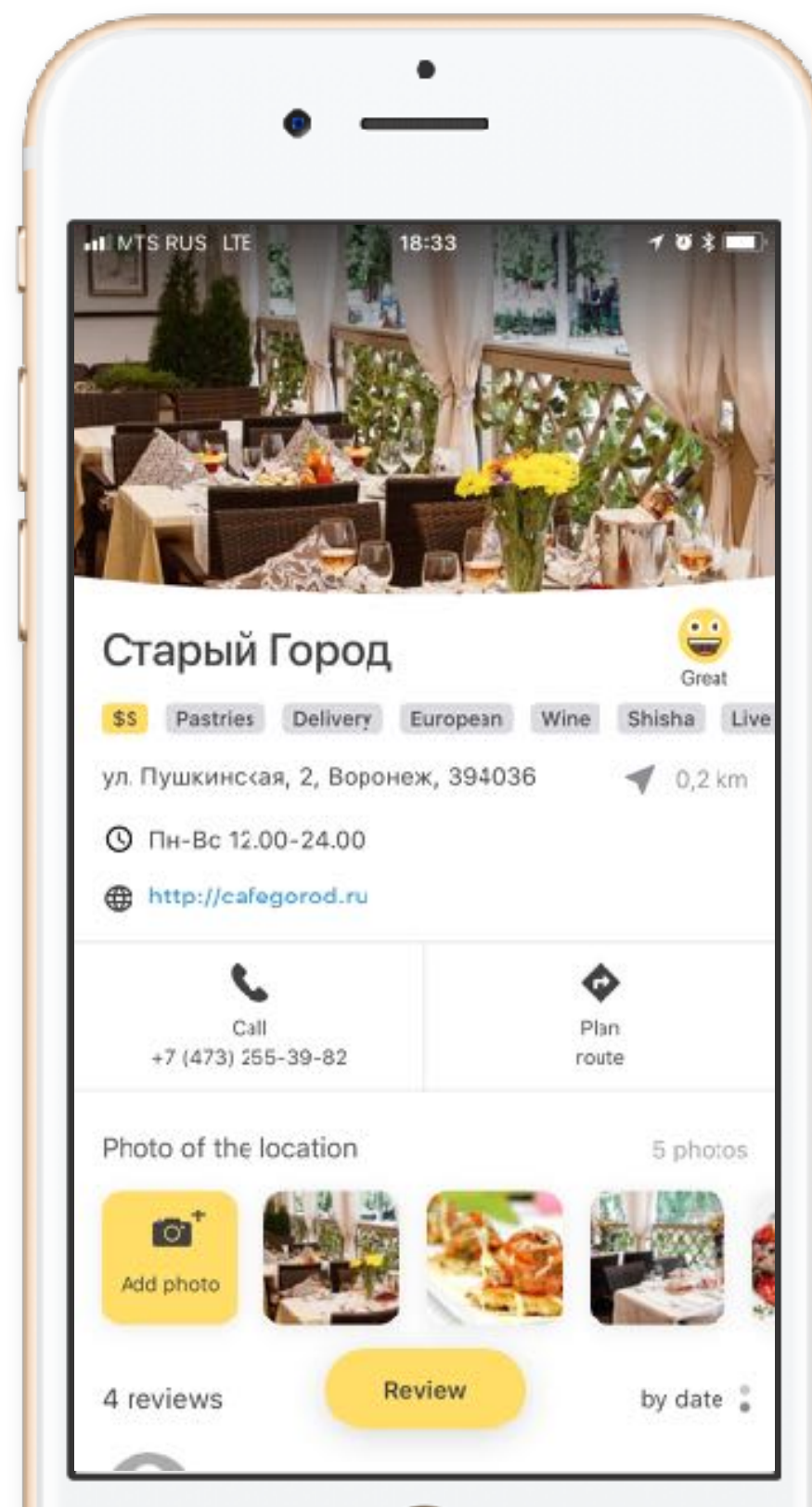
Review

by date



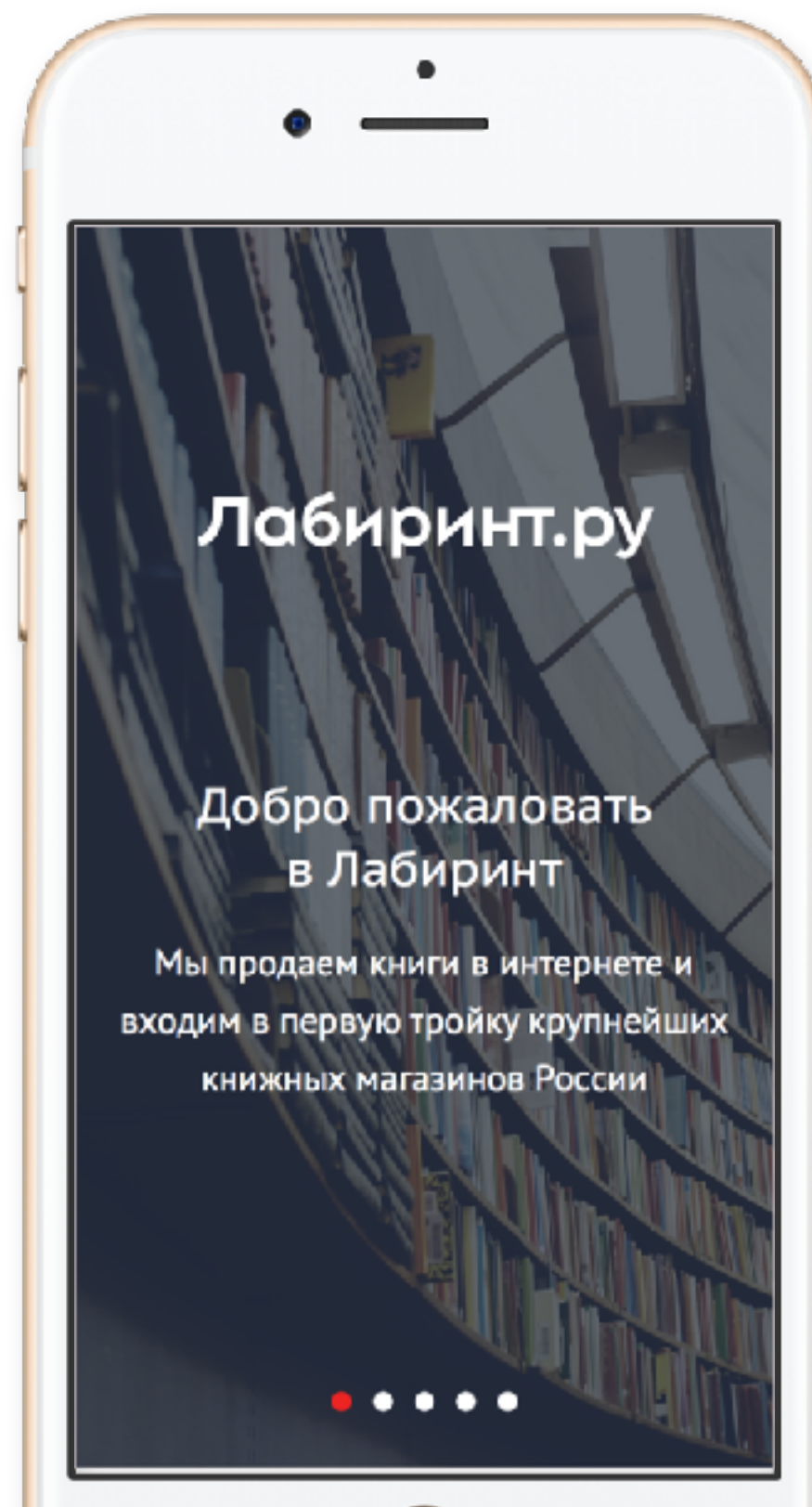
Виды данных

- 🍏 **Контент**
товары, тексты, фильмы, книги, ...
- 🖼️ **Медиа**
картинки, гифки, музыка, ...
- 🔑 **Приватные данные**
токены, пароли, номера карт, ...
- 📦 **Общие данные**
настройки, флаги, API ключи, ...





Виды данных

- 🍏 **Контент**
товары, тексты, фильмы, книги, ...
- 🖼️ **Медиа**
картинки, гифки, музыка, ...
- 🔑 **Приватные данные**
токены, пароли, номера карт, ...
- 📦 **Общие данные**
настройки, флаги, API ключи, ...



Виды данных

-  **Контент**
товары, тексты, фильмы, книги, ...
-  **Медиа**
картинки, гифки, музыка, ...
-  **Приватные данные**
токены, пароли, номера карт, ...
-  **Общие данные**
настройки, флаги, API ключи, ...



Данные в приложениях

Работа с данными

Хранение данных

Данные в приложениях

Работа с данными

Хранение данных

Работа с данными

- Откуда и как брать?
- Как обрабатывать?
- Как отправлять?

Откуда получать

- 🧑 Пользователь
- 🌐 Интернет
- 📱 Устройство
- ➡️📱 Входящие данные

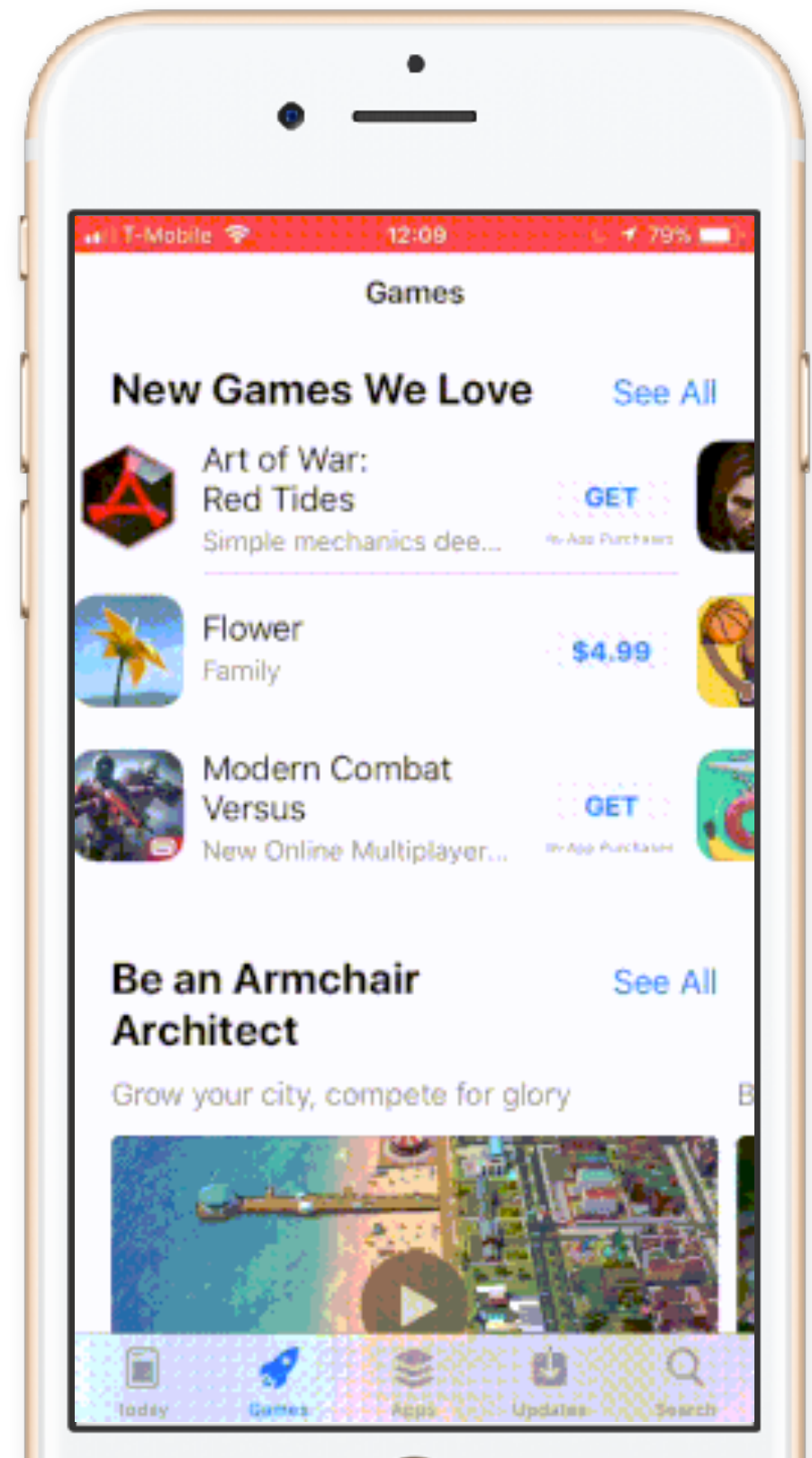
Откуда получать

- 😊 **Пользователь**
- 🌐 **Интернет**
- 📱 **Устройство**
- ➡️📱 **Входящие данные**



Откуда получать

- 😊 Пользователь
- 🌐 Интернет
- 📱 Устройство
- ➡️ 📱 Входящие данные



Откуда получать

- 😊 Пользователь
- 🌐 Интернет
- 📱 Устройство
- ➡️ 📱 Входящие данные



Откуда получать

- 😊 Пользователь
- 🌐 Интернет
- 📱 Устройство
- ➡️ 📱 Входящие данные



От пользователя

- Поля ввода и контролы
UITextField, UITextView
- Буфер обмена
UIPasteboard
- Камера, микрофон
AVFoundation
- “Открыть в...”
Определить UTI в своем приложении

Из интернета



Из интернета

- Работа с сетью в iOS
- Сериализация/десериализация
- Типы API
- Библиотеки
- Архитектура сетевого слоя
- Инструменты

Работа с сетью в iOS

```
let task = URLSession.shared.dataTask(with: url) {(data, response, error) in

    guard error == nil else {
        print("returning error")
        return
    }

    guard let content = data else {
        print("not returning data")
        return
    }

    guard let json = (
        try? JSONSerialization.jsonObject(
            with: content,
            options: JSONSerialization.ReadingOptions.mutableContainers
        )
    ) as? [String: Any] else {
        print("Not containing JSON")
        return
    }

    // Process JSON

}
task.resume()
```

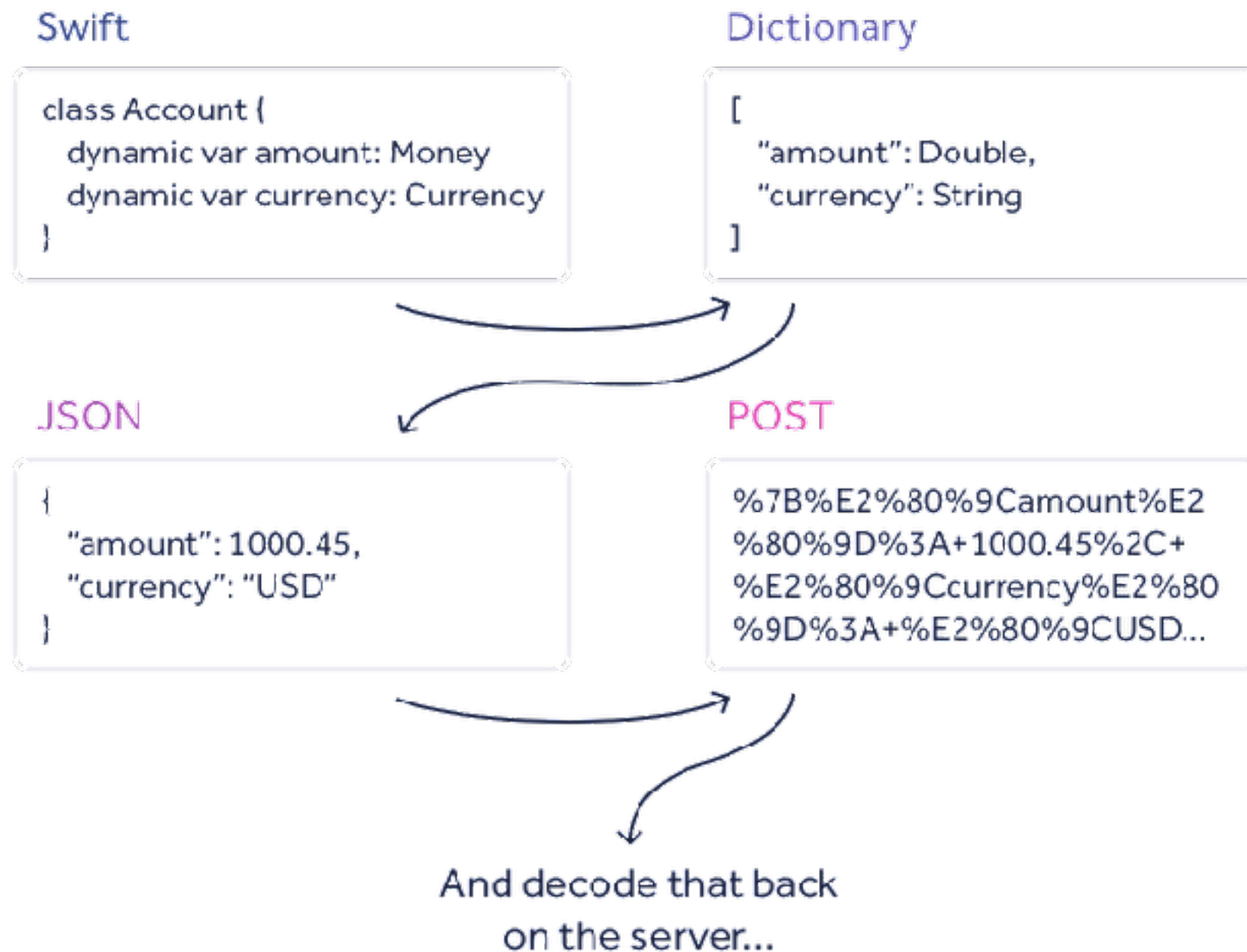
Типы API

- REST
- SOAP
- WebSockets
- gRPC
- ...

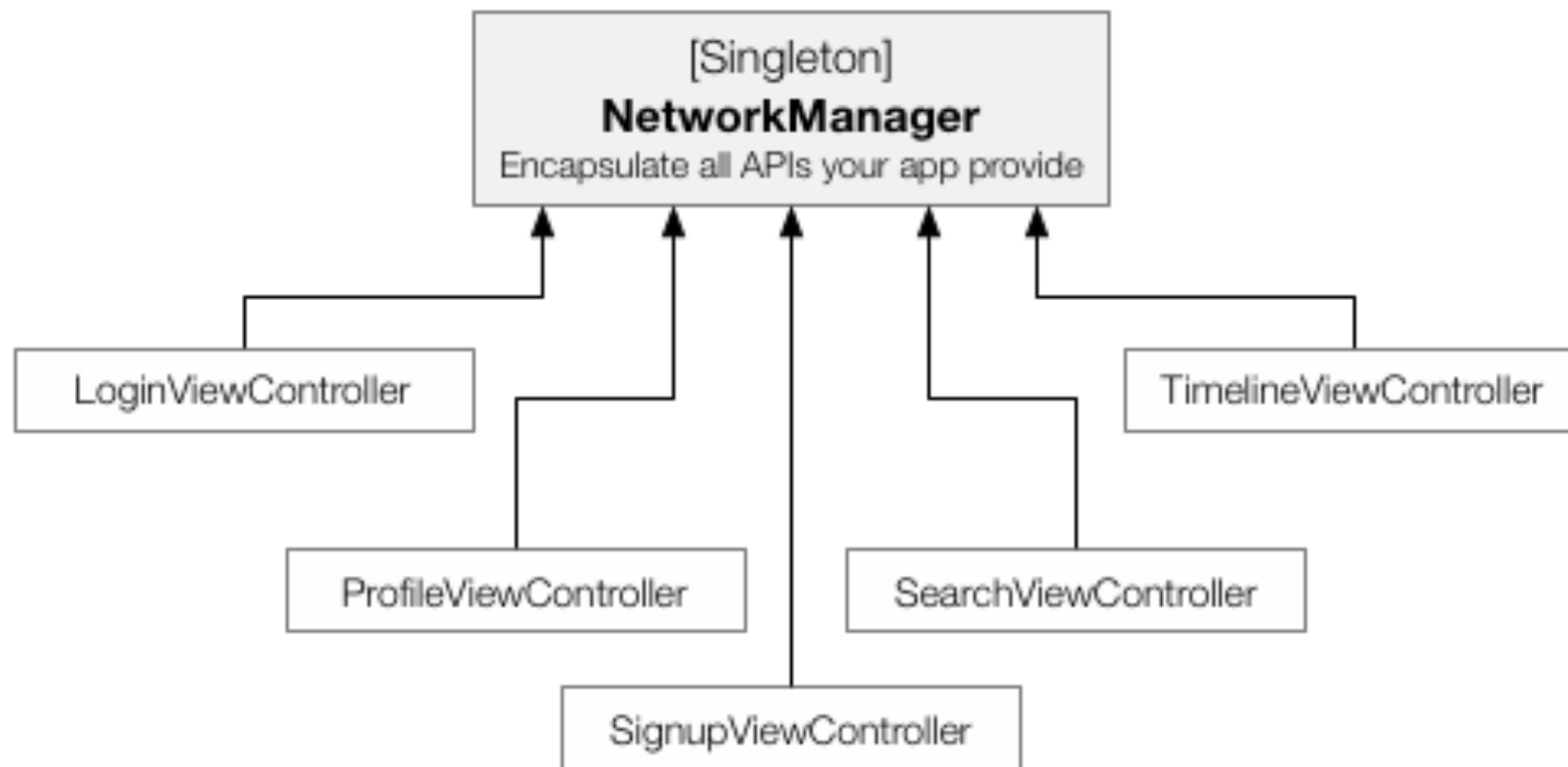
Типы API

- **REST**
- SOAP
- **WebSockets**
- gRPC
- ...

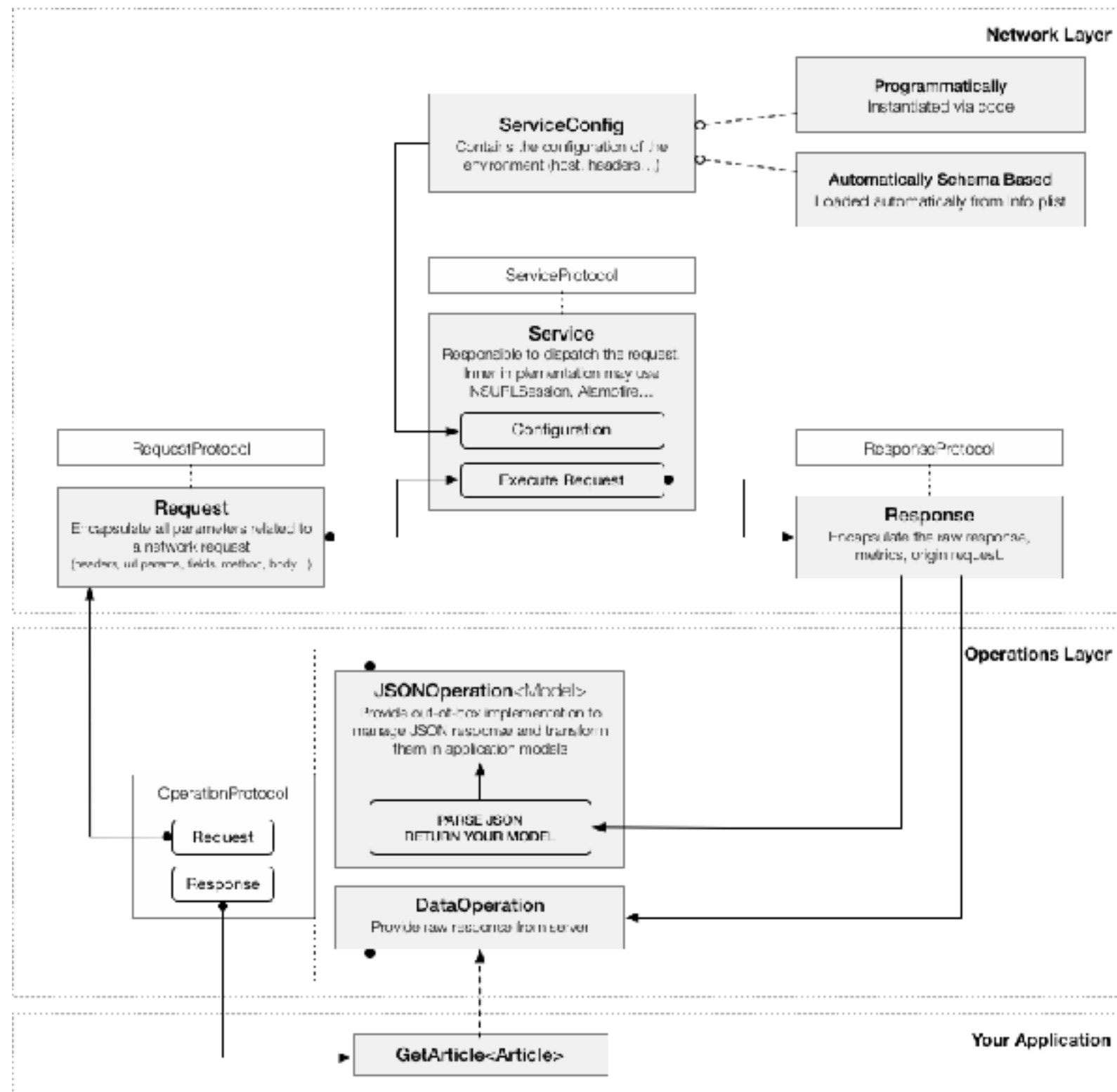
Сериализация



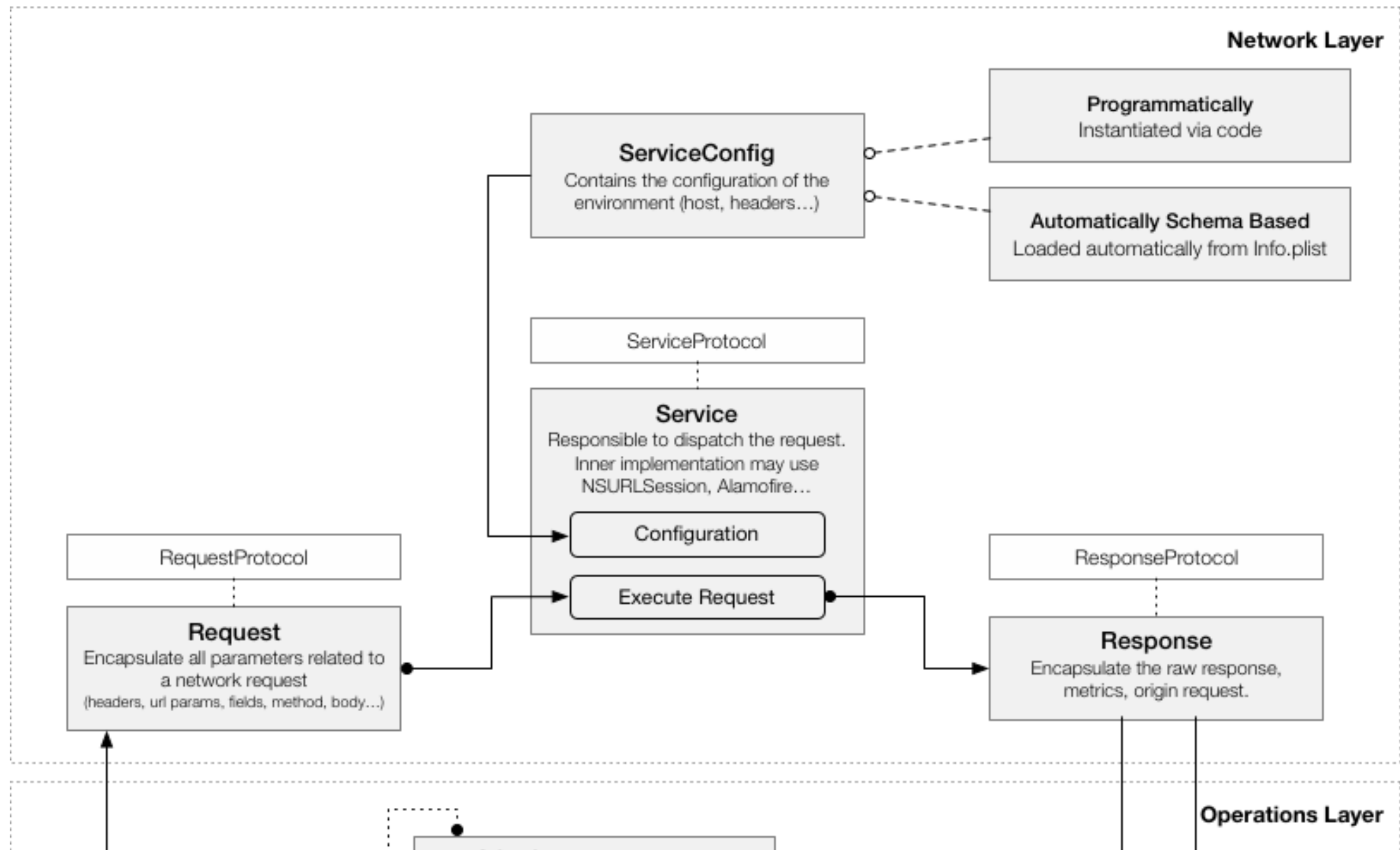
Сетевой слой



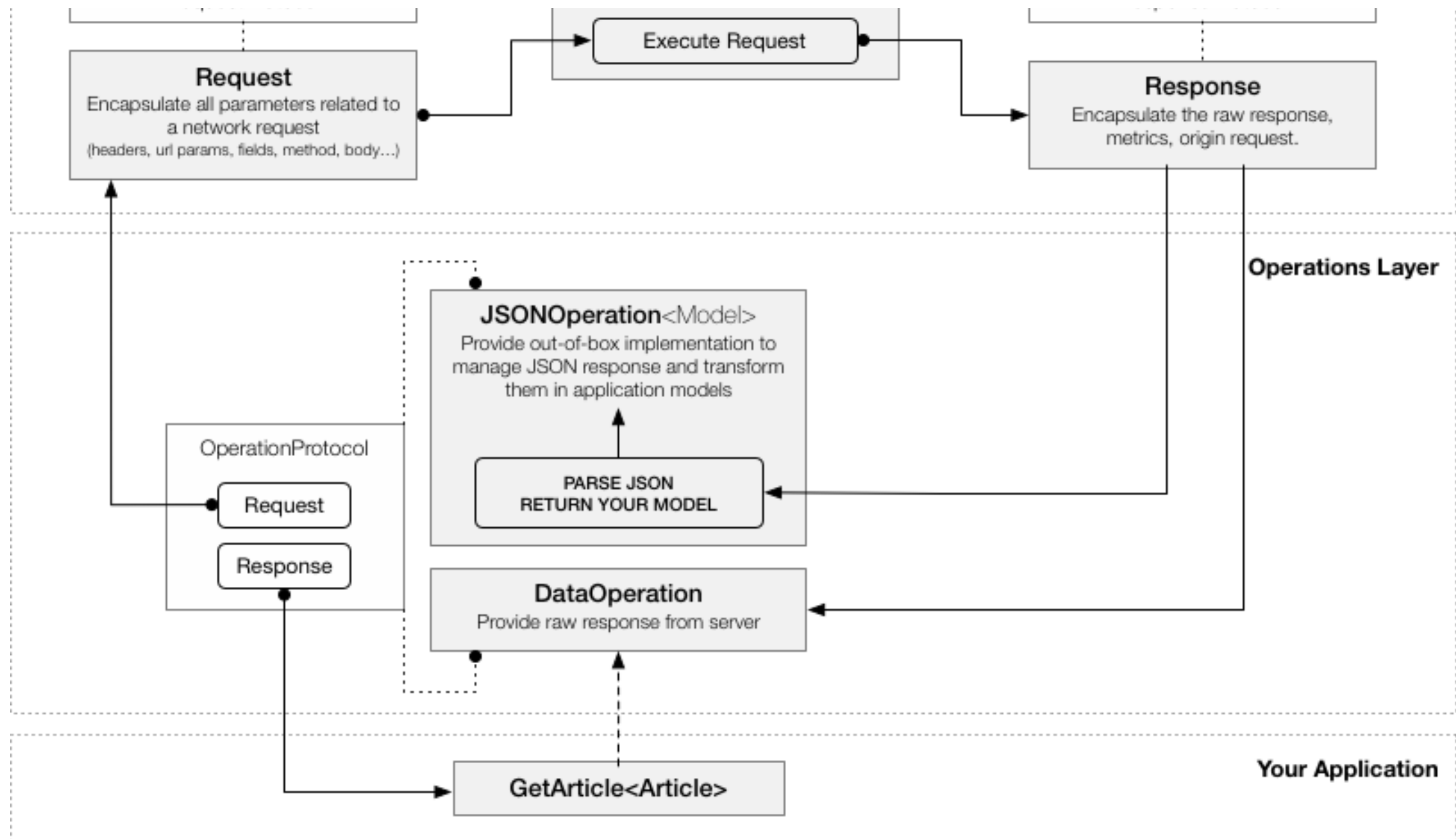
Сетевой слой



Сетевой слой



Сетевой слой



Библиотеки

- 🍰 Alamofire / SwiftyHTTP
- 🖼️ Alamofire+Image / Nuke
- 💎 Moya

// Moya example

```
struct Repo {  
    let url: URL  
    let title: String  
}
```

```
extension Repo: Decodable {  
    // ...  
}
```

```
struct RepoResults {  
    let page: Int  
    let numberOfResults: Int  
    let nuberoOfPages: Int  
    let repos: [Repo]  
}
```

```
extension RepoResults: Decodable {  
    // ...  
}
```

```
enum ReposApi {  
    case popular(page: Int)  
    case newRepos(page: Int)  
    case repo(id: Int)  
}
```

Инструменты

- POSTMAN
- Paw
- Charles

Из устройства

- События системы и системная информация
Core Services, Core Telephony, CoreWLAN, ...
- Системные фреймворки
AddressBook, CallKit, MapKit, Speech, ...

Входящие данные

- Push Notifications
- DeepLinks
 - UniversalLinking, DynamicLinks, ...*
- Handoff
- “Открыть в...”
 - UTI*

Отправка данных

- Отправка на сервер
- Share
- Буфер обмена
- Обмен данными между приложениями

Демо

Данные в приложениях

Работа с данными

Хранение данных

Данные в приложениях

Работа с данными

Хранение данных

Где хранить

- Оперативная память
- Файловая система
- UserDefaults
- Keychain
- База данных

Файловая система

- Documents
- Library
 - Caches
 - ...
- tmp

Файловая система

Работа с директориями

```
// Documents  
let documentsPath = FileManager.default.urls(  
    .documentDirectory, // .cachesDirectory, ...  
    .userDomainMask  
).first
```


Файловая система

Сохранение данных

```
...  
  
let string = "Hello world"  
let filename = documentsPath.appendingPathComponent("test.txt")  
  
do {  
    try string.write(  
        to: filename,  
        atomically: true,  
        encoding: String.Encoding.utf8  
    )  
} catch {  
    // Error  
}
```

Файловая система

Хранение объектов

```
struct Person: Codable {  
    let name: String  
    let age: Int  
    ...  
}  
  
let person = Person(name: "Mark", age: 25)  
  
// Saving  
NSKeyedArchiver.archiveRootObject(person, toFile: filename)  
  
//Loading  
guard let person = NSKeyedUnarchiver.unarchiveObject(  
    withFile: filename  
) as? Person else { return }
```

UserDefaults

```
// Saving  
UserDefaults.standard.set("Gregory", forKey: "name")
```

```
// Loading  
let name = UserDefaults.standard.string(  
    forKey: "name"  
) ?? ""
```

Keychain

```
let key = <# a key #>  
let tag = "com.example.keys.mykey".data(using: .utf8)!  
let addquery: [String: Any] = [kSecClass as String: kSecClassKey,  
                               kSecAttrApplicationTag as String: tag,  
                               kSecValueRef as String: key]  
  
...  
  
let status = SecItemAdd(addquery as CFDictionary, nil)  
guard status == errSecSuccess else { throw <# an error #> }  
  
...  
  
let getquery: [String: Any] = [kSecClass as String: kSecClassKey,  
                                kSecAttrApplicationTag as String: tag,  
                                kSecAttrKeyType as String: kSecAttrKeyTypeRSA,  
                                kSecReturnRef as String: true]  
  
...
```

Keychain

<https://github.com/kishikawakatsumi/KeychainAccess>

```
let keychain = Keychain(service: "com.example.keychain.app")  
keychain["myapikey"] = "<API KEY>"
```

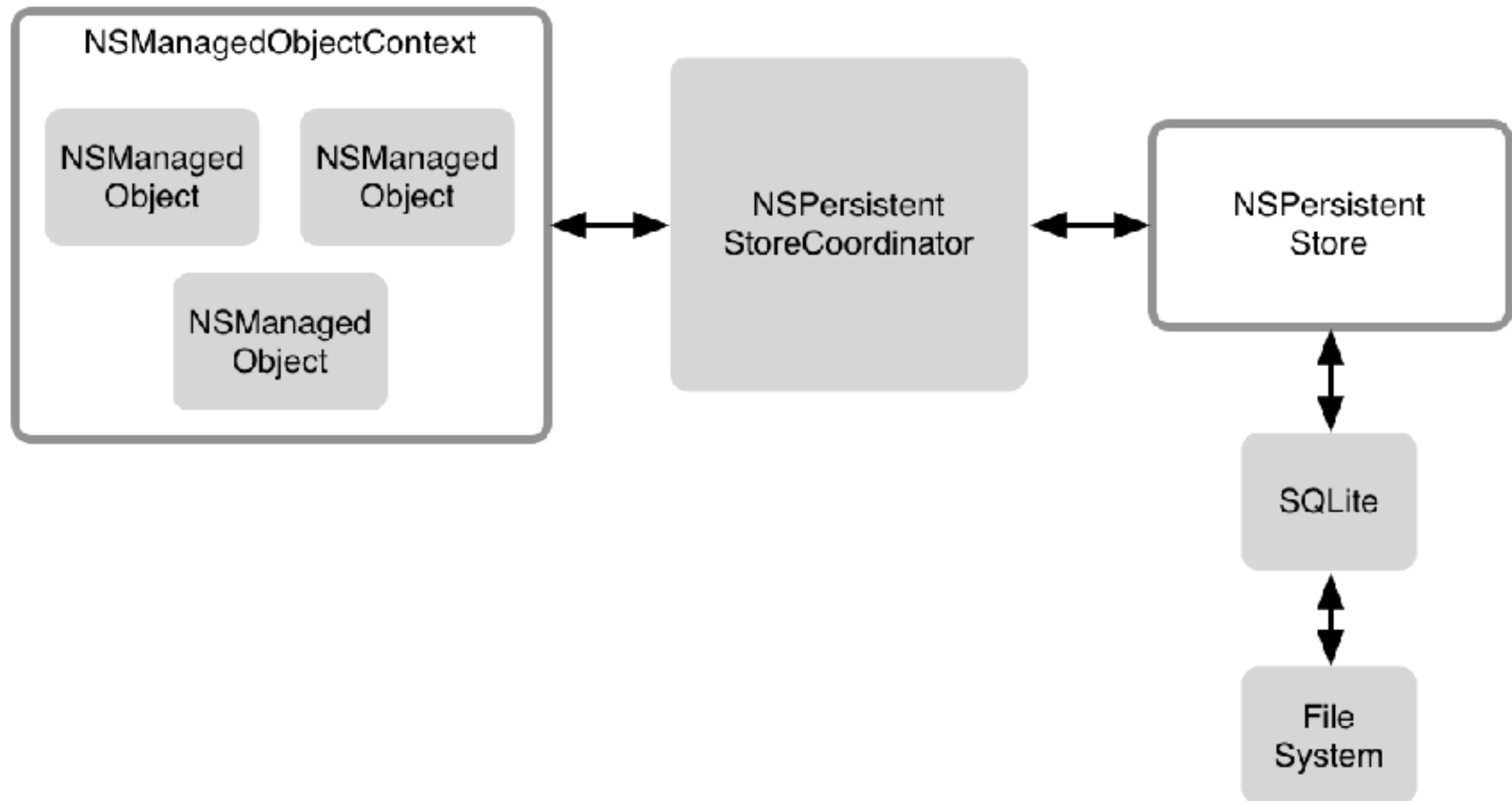
```
...
```

```
let token = keychain["myapikey"]
```

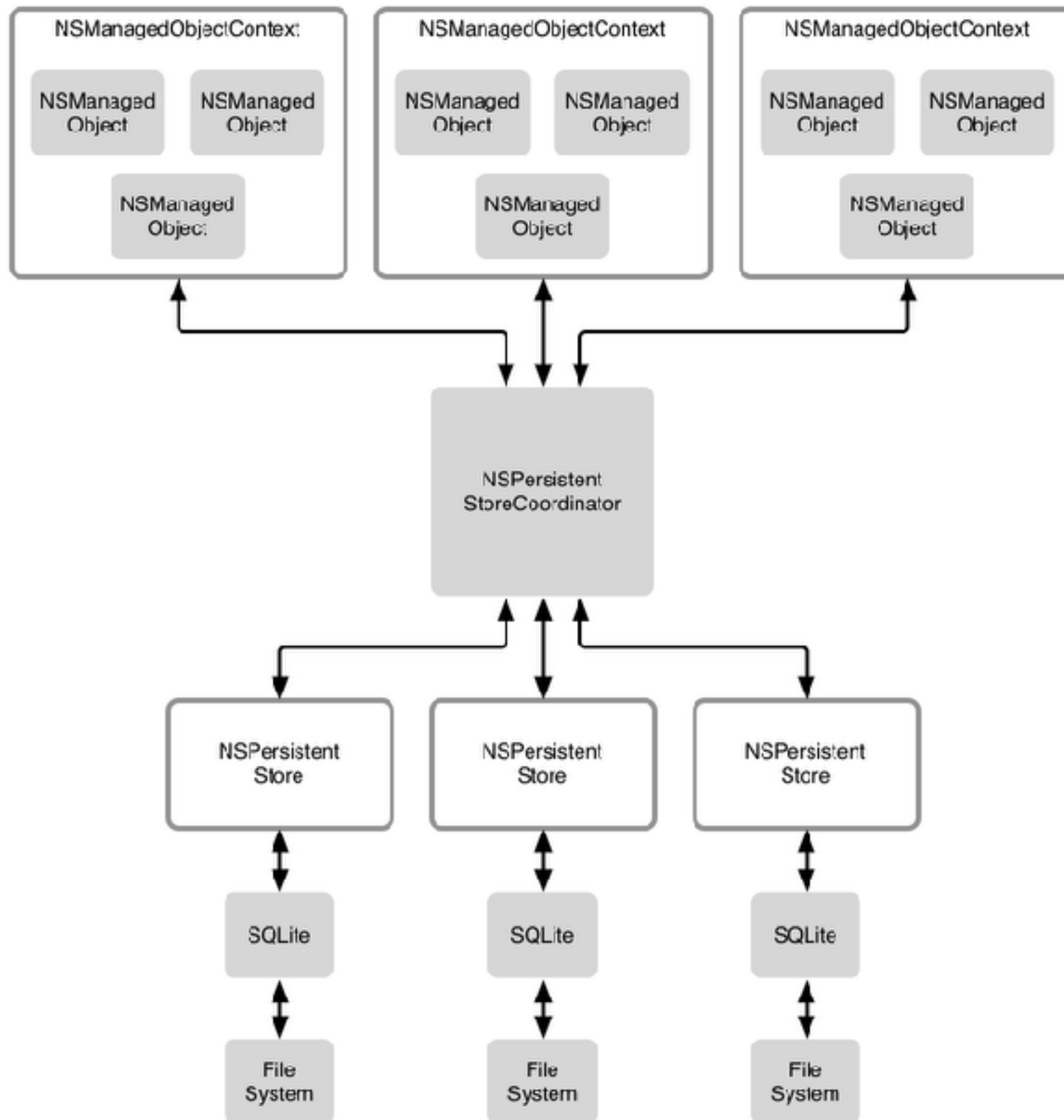
База данных

- CoreData
- Sqlite
- Realm
- ...

Core Data



Core Data



Core Data

- Apple
- Язык запросов
- Частичная подгрузка данных
- Трекинг изменений
- Возможность отменять действия
- Каскадное удаление
- Контексты
- ...

Демо

Realm

- Простая в использовании
- Не потокобезопасна
- Нет каскадного удаления
- Быстрее
- Кроссплатформенна

Данные в приложениях

Работа с данными

Хранение данных

Данные в приложениях

Работа с данными

Хранение данных

Материалы

- [https://github.com/surfstudio/iOSSummerSchool2018/blob/master/Лекция 5. Работа с данными/README.md](https://github.com/surfstudio/iOSSummerSchool2018/blob/master/Лекция%205.%20Работа%20с%20данными/README.md)

