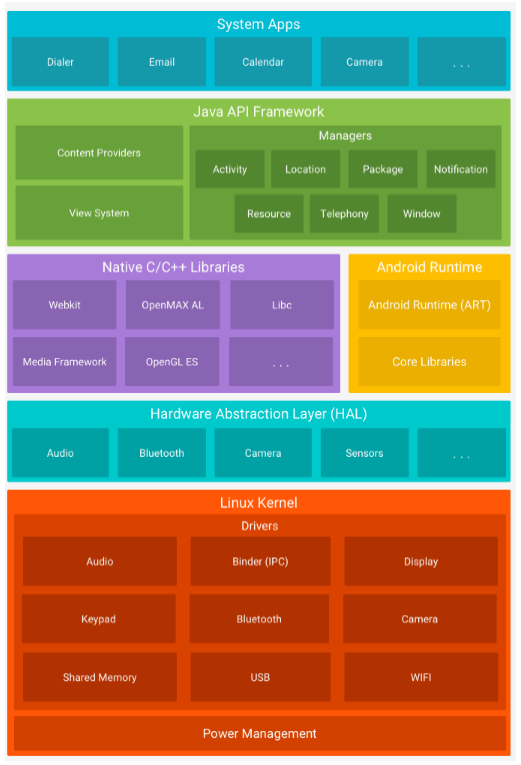
**1. Как происходит процесс построения приложения?**

* считывание AndroidManifest.xml
* Затем вызывается программа aapt, она проверяет ресурсы и компилирует их, создавая при этом класс R.java в котором содержатся идентификаторы ресурсов и файл resources.arsc в котором содержится информация об xml-ресурсах и их атрибутах.
* Далее подхватываются все библиотеки, которые используются в проекте и запускается Java-компилятор javac.
* Полученные class-файлы передаются в программу dx, которая переводит их в dex-формат. Причём для оптимизации, готовые библиотеки дексируются отдельно, а классы проекта отдельно (оптимизация в том, что дексированные библиотеки можно закешировать).
* Если собралось несколько dex-файлов, то они все объединяются при помощи Dex Merger Tool. В конечном итоге мы получаем единственный файл classes.dex (или несколько, если используется multidex).
* Теперь у нас есть все компоненты и можно собирать apk.

**2. Сравните нативные и кроссплатформенные приложения.**

* **Нативные приложения** разработаны специально под конкретную платформу (например, Android или iOS) с использованием соответствующих языков программирования и инструментов разработки.
* **Кроссплатформенные приложения** разработаны с использованием фреймворков, позволяющих создавать приложения для нескольких платформ одновременно. Это может быть через использование языков, таких как React Native, Flutter, Xamarin, или через веб-технологии (например, с использованием Cordova или Ionic).

**3. Опишите архитектуру Android.**

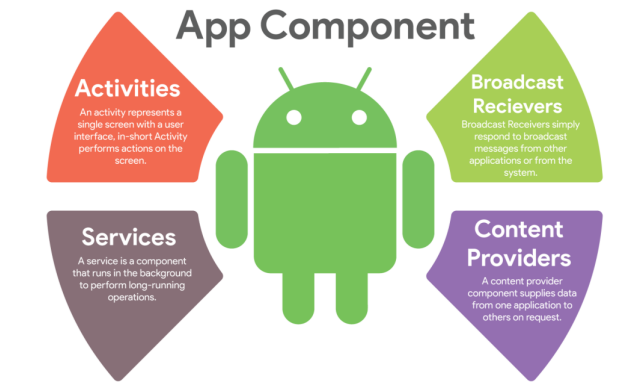


**4. Что такое ART?**

Начиная с версии Android 4.4. KitKat в ОС Android была добавлена вторая виртуальная машина - Android RunTime. Хотя в той же версии Android 4.4 Dalvik по умолчанию является предпочтительной средой для выполнения кода приложения, но уже в версии Android 5.0 Lollipop виртуальная машина Android RunTime полностью заменила Dalvik.

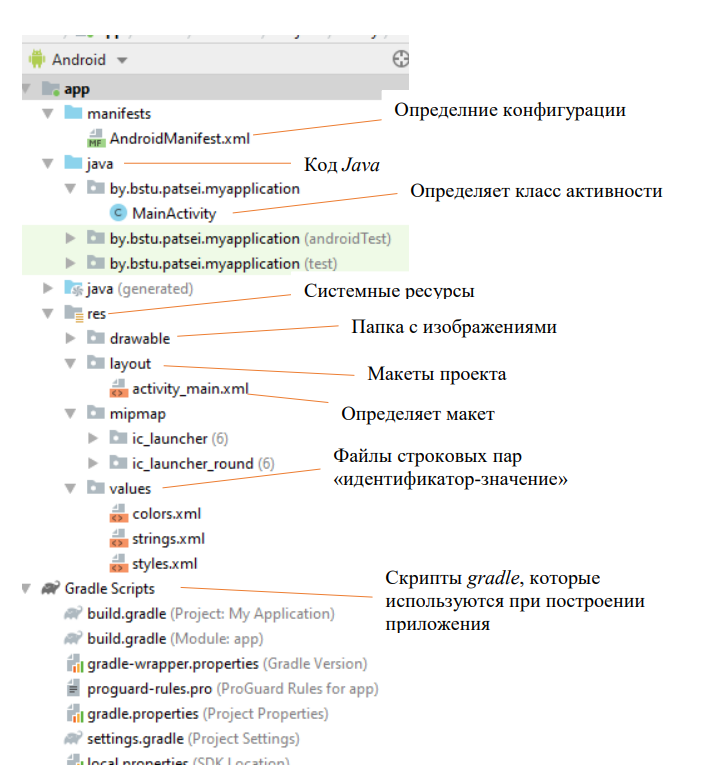
Работа Android RunTime коренным образом отличается от Dalvik: Android RunTime использует предварительную компиляцию приложения при его установке на мобильное устройство, а не JIT-компиляцию, что увеличивает производительность приложения.

**5. Какие компоненты может содержать приложение? Дайте им характеристику.**



* **Activities**: Представляют пользовательский интерфейс.
* **Services**: Выполняют операции в фоновом режиме.
* **Broadcast Receivers**: Отвечают на трансляции системных событий.
* **Content Providers**: Управляют доступом к данным приложения.

**6. Опишите структуру проекта приложения для Android.**



**7. Как организованы ресурсы проекта? Перечислеите, папки и формат.**

Ресурсы в проекте Android организованы в папке res, которая содержит подпапки для различных видов ресурсов, таких как макеты (layout), изображения (drawable), строки (values) и другие. Форматы ресурсов могут быть XML для разметки и других настроек, а также изображения в форматах PNG, JPEG и других.

**8. Как обратиться к ресурсу?**

Для доступа к ресурсам из кода используются методы из классов R (генерируется автоматически) в Java или @drawable, @string, @layout в XML.

**9. Что такое манифест приложения и что он описывает.**

AndroidManifest.xml - это дескриптор приложения, который содержит информацию о приложении, такую как разрешения, компоненты приложения (Activities, Services), минимальные версии SDK, версию приложения и другие метаданные, необходимые для запуска приложения и взаимодействия с операционной системой Android.