

vivid

GRADUS

На

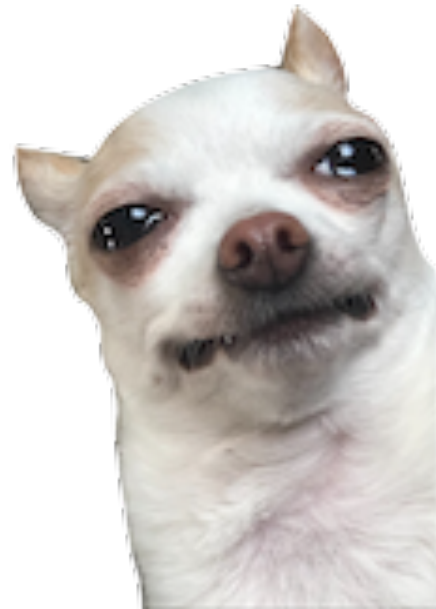
прокачку

vivid Money



- Super App для финансовых сервисов (Европа)
- Mobile only
- 15+ разработчиков на каждую мобильную платформу

Специальная нотация



– Basic



– Intermediate



– Advanced

Оглавление

- Структура проекта   
- Build cache 
- Annotation processing   
- buildSrc   
- Отключаем фичи AGP 
- Jetifier  

Основные моменты (Июнь 2020)

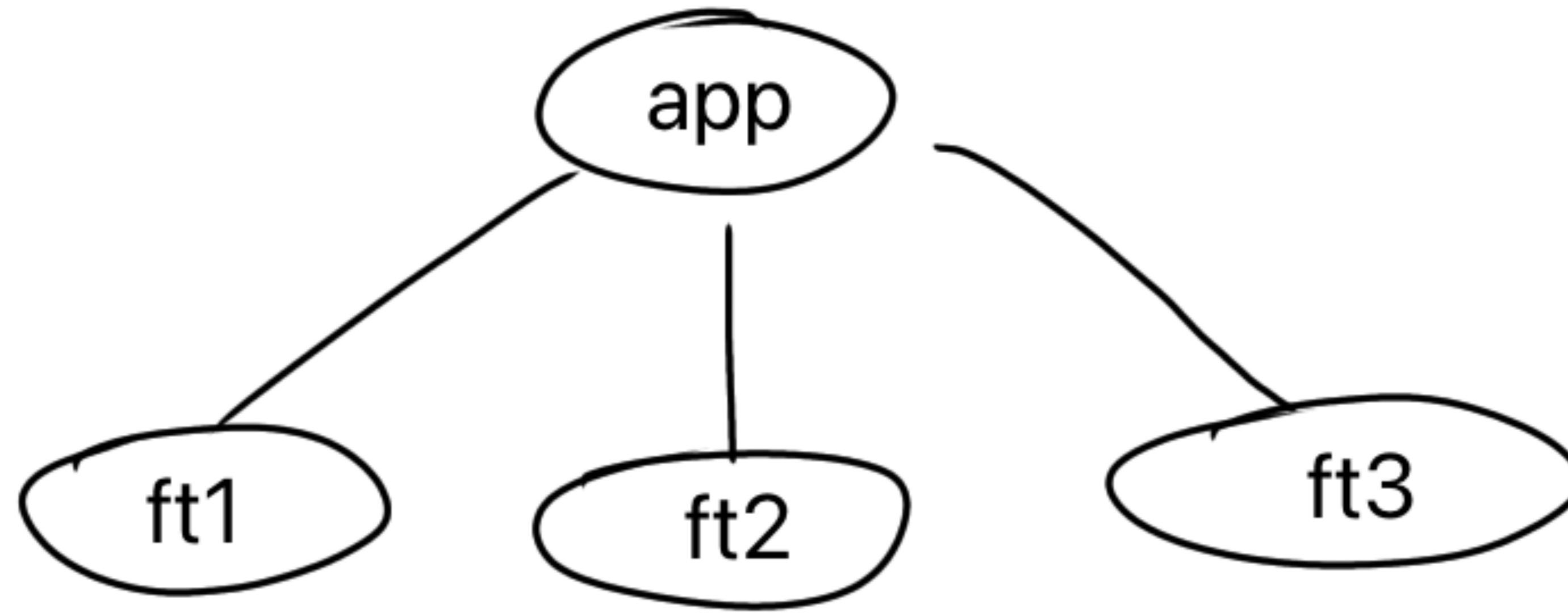
- KLOC: **~150k**
- Gradle модули: **~100**
- **buildSrc**
- Чистый билд: **~7 min**
- Инкрементальный билд: **~3 min**





Структура проекта

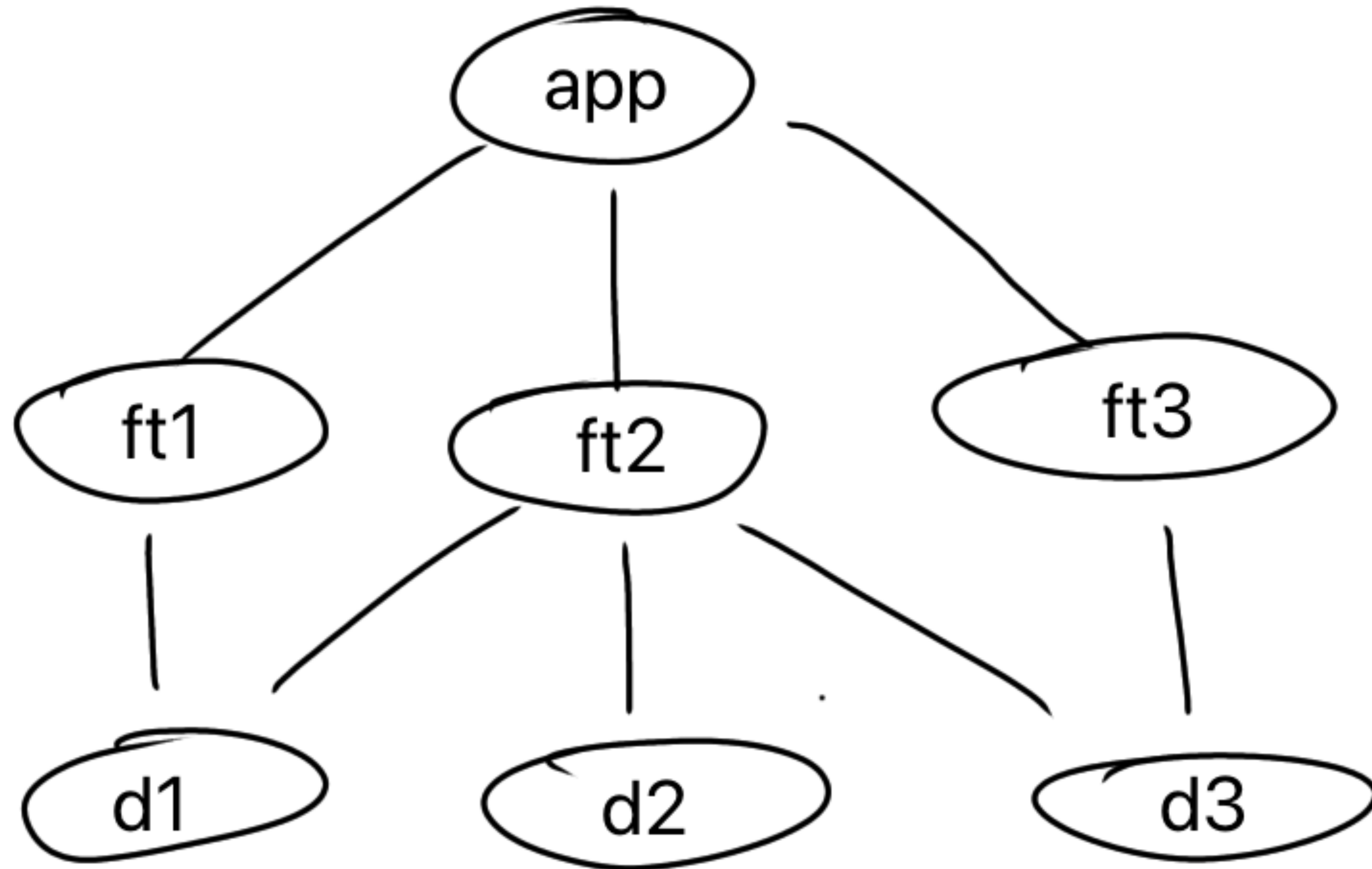
app



Feature модуль

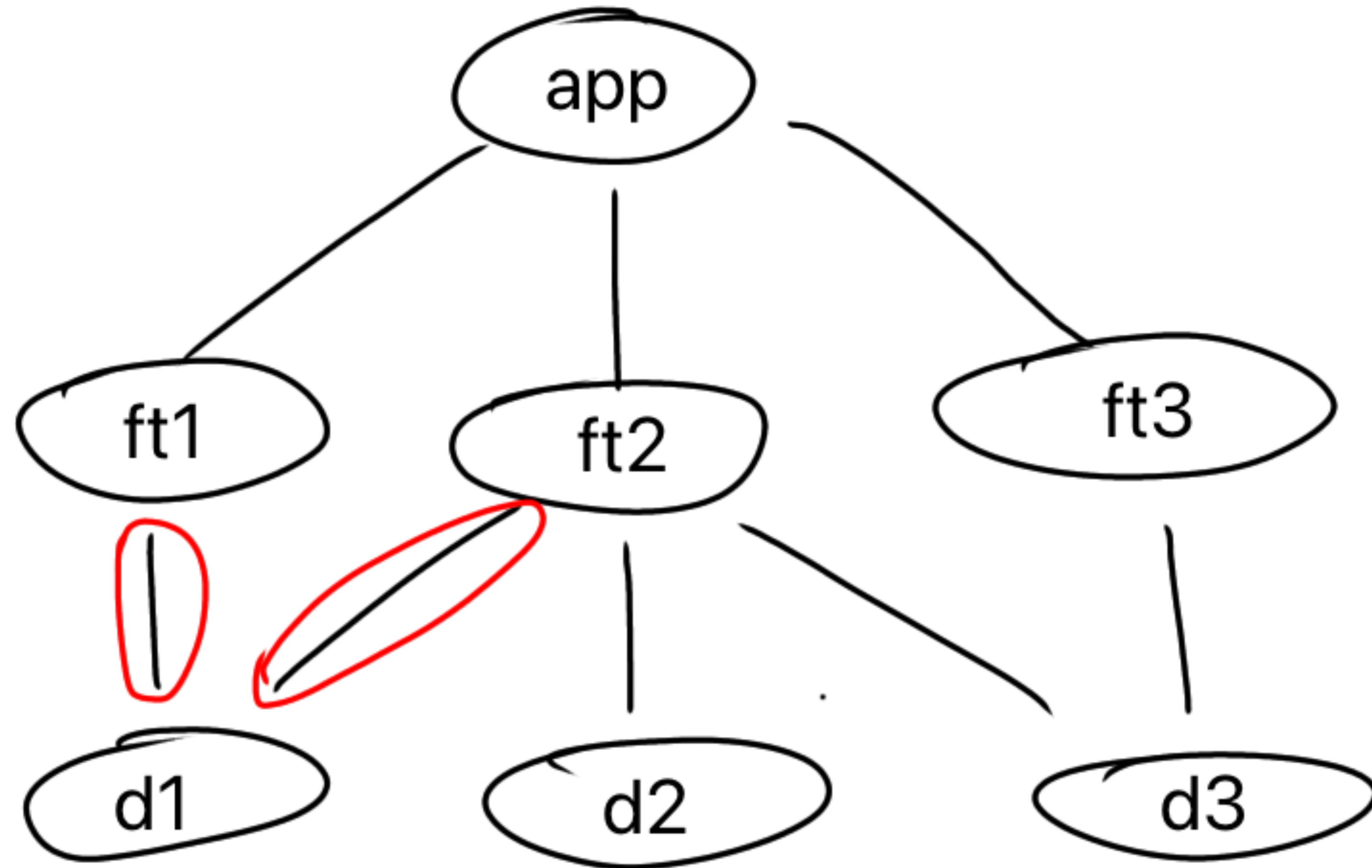
- View (Fragment)
- MVI сущности
- DI

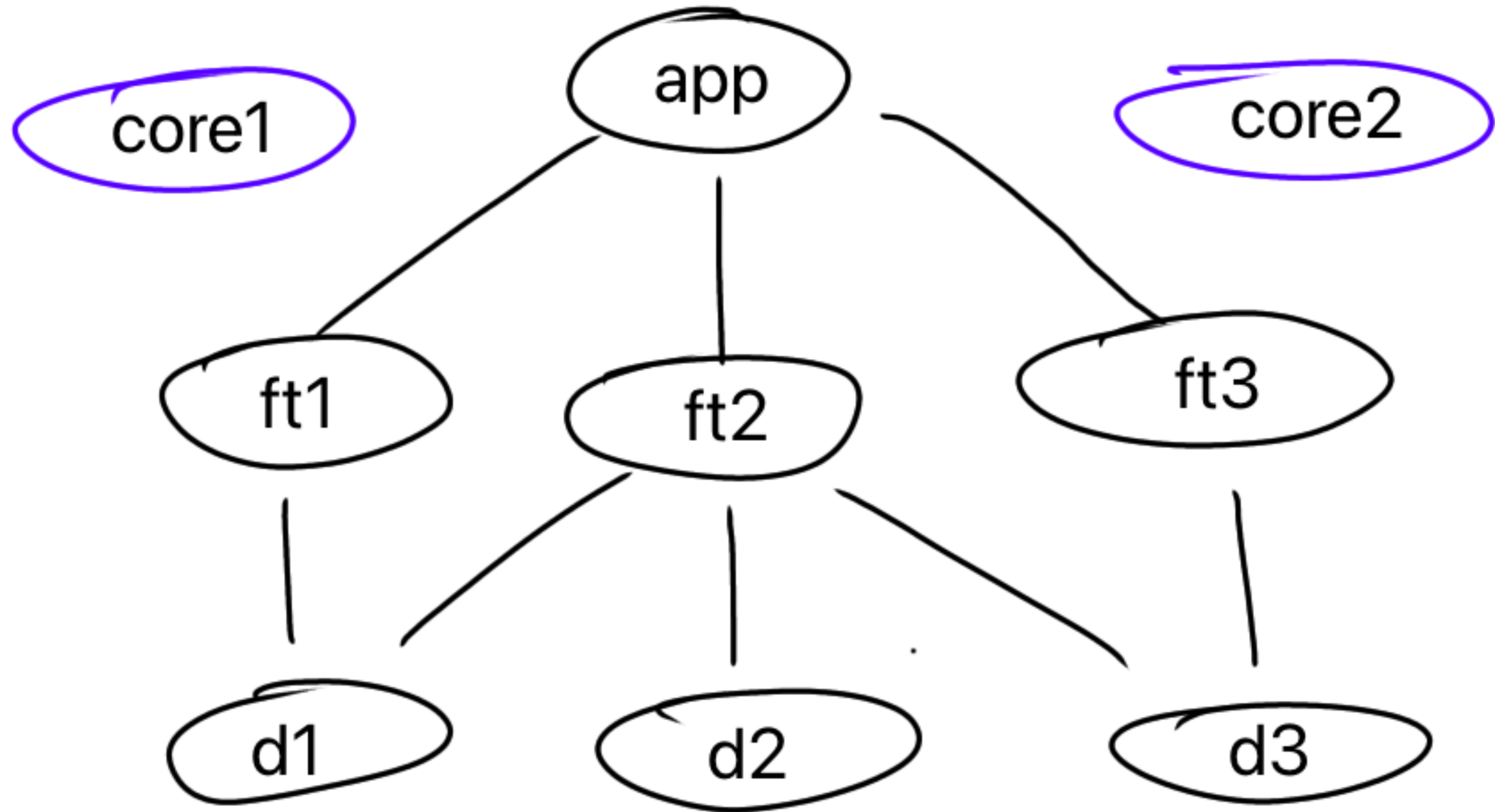
org.gradle.parallel=true



Domain модуль

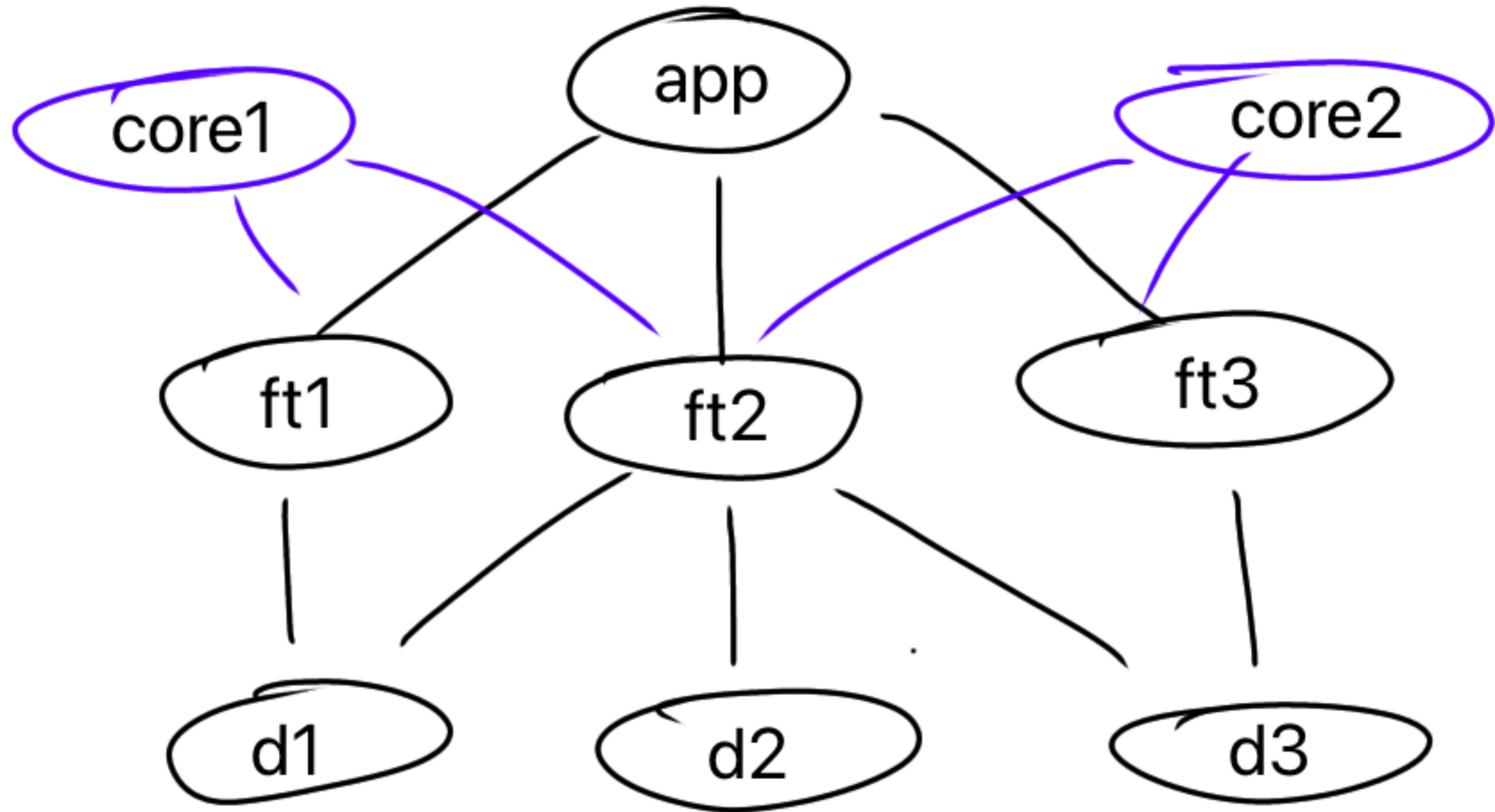
- Domain модели
- Репозитории
- API фичи
- **Чистый Kotlin модуль**

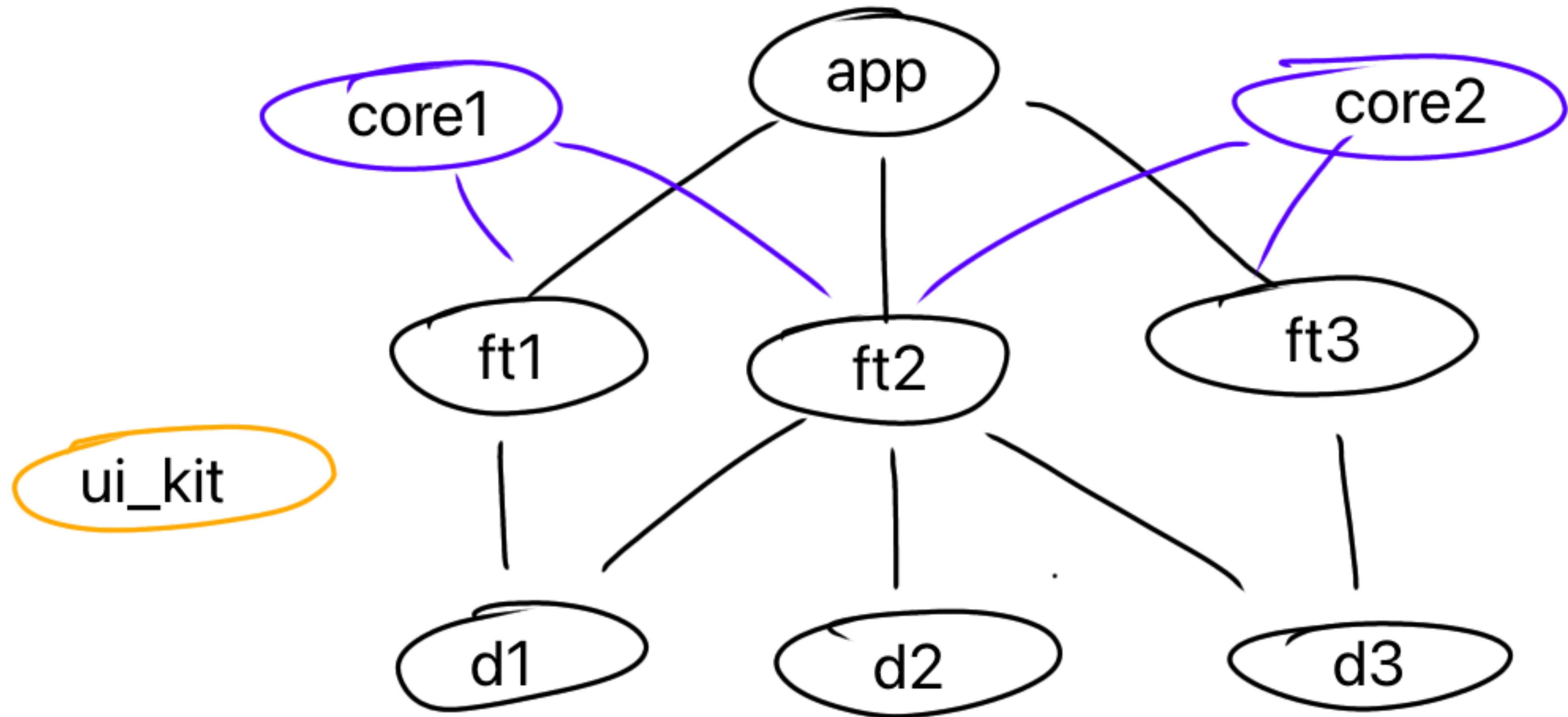


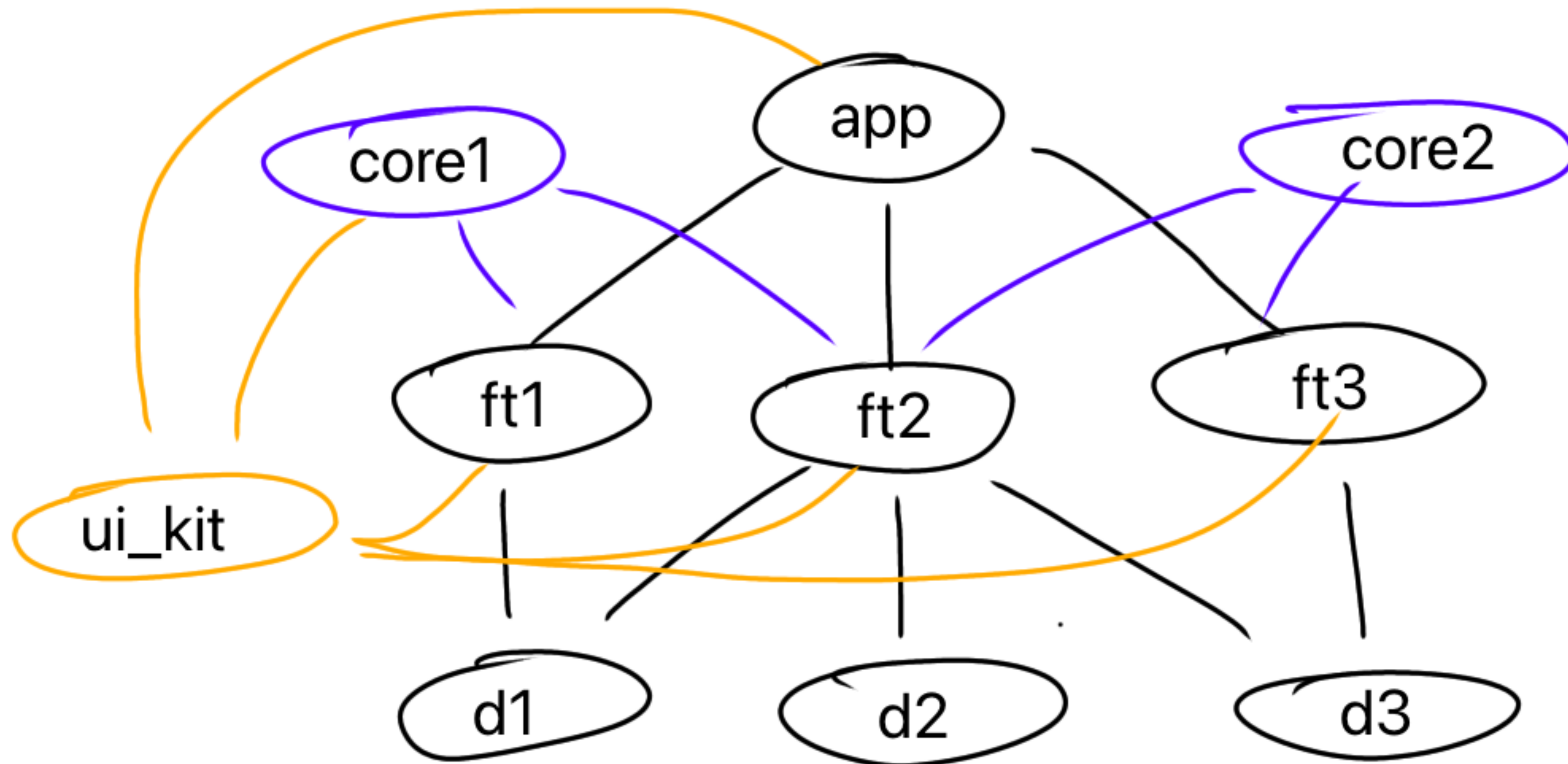


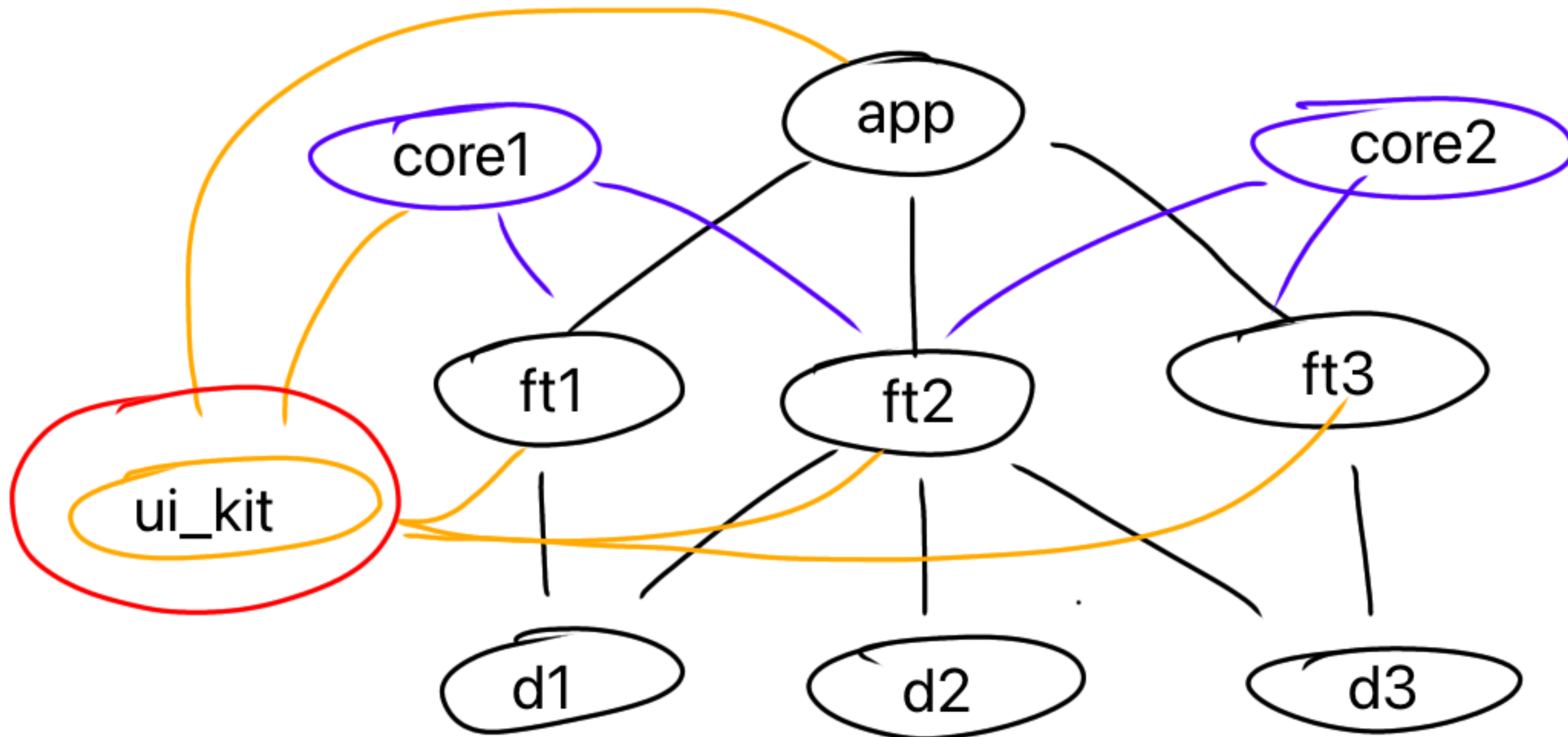
Core модуль

- Network
- MVI
- Cache
- Аналитика
- etc.
- **Предпочтительно чистый Kotlin модуль**





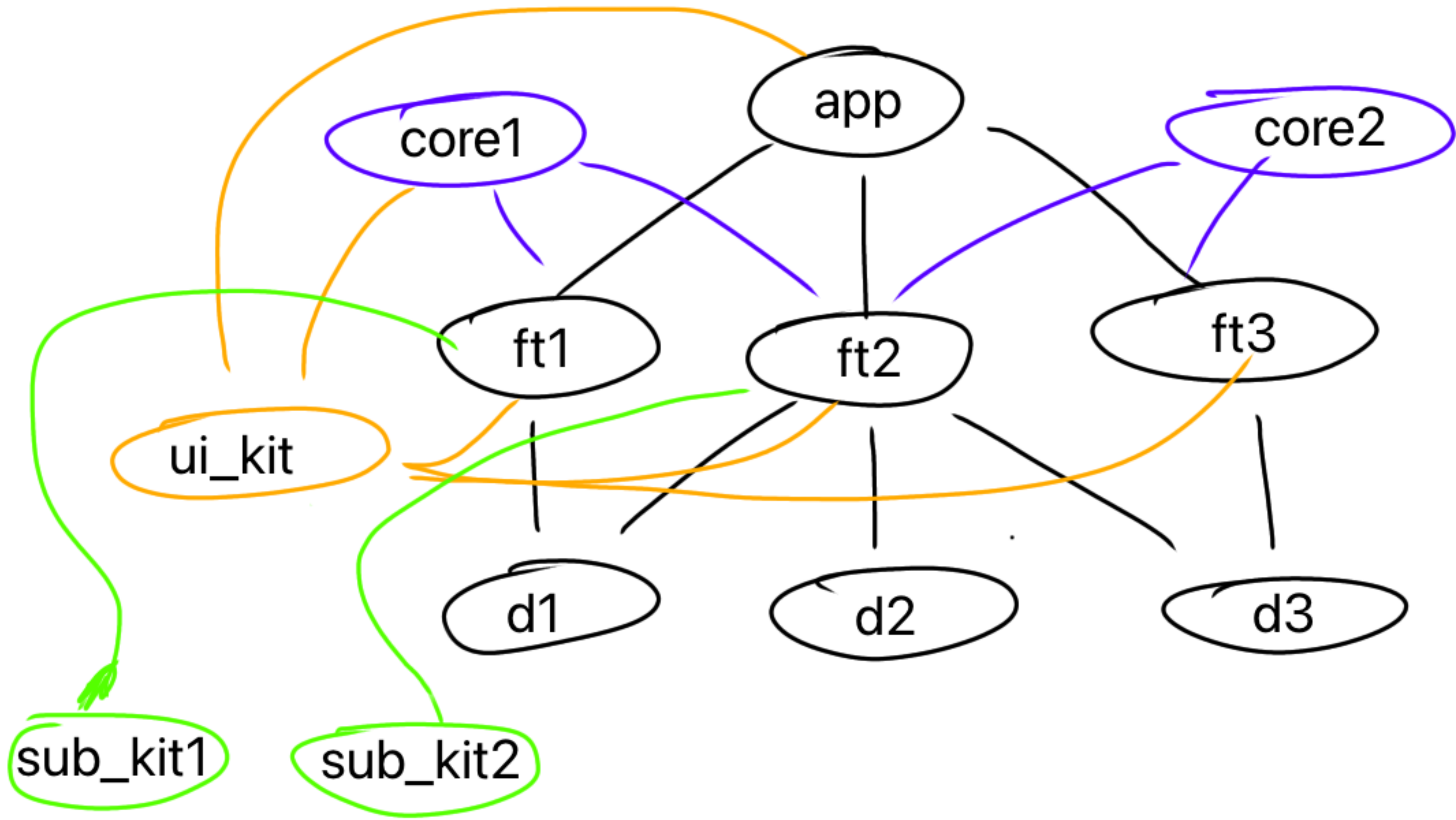


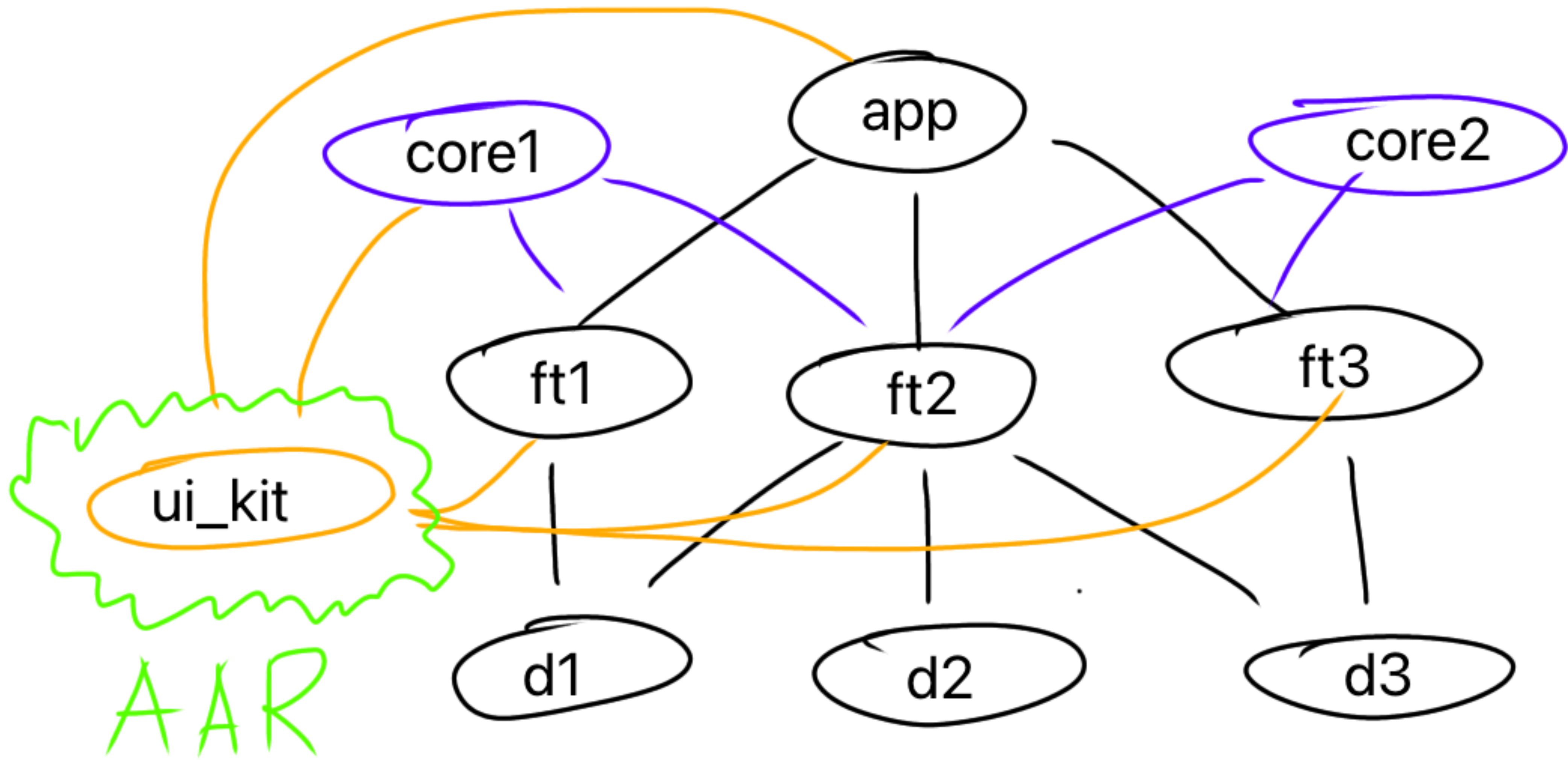


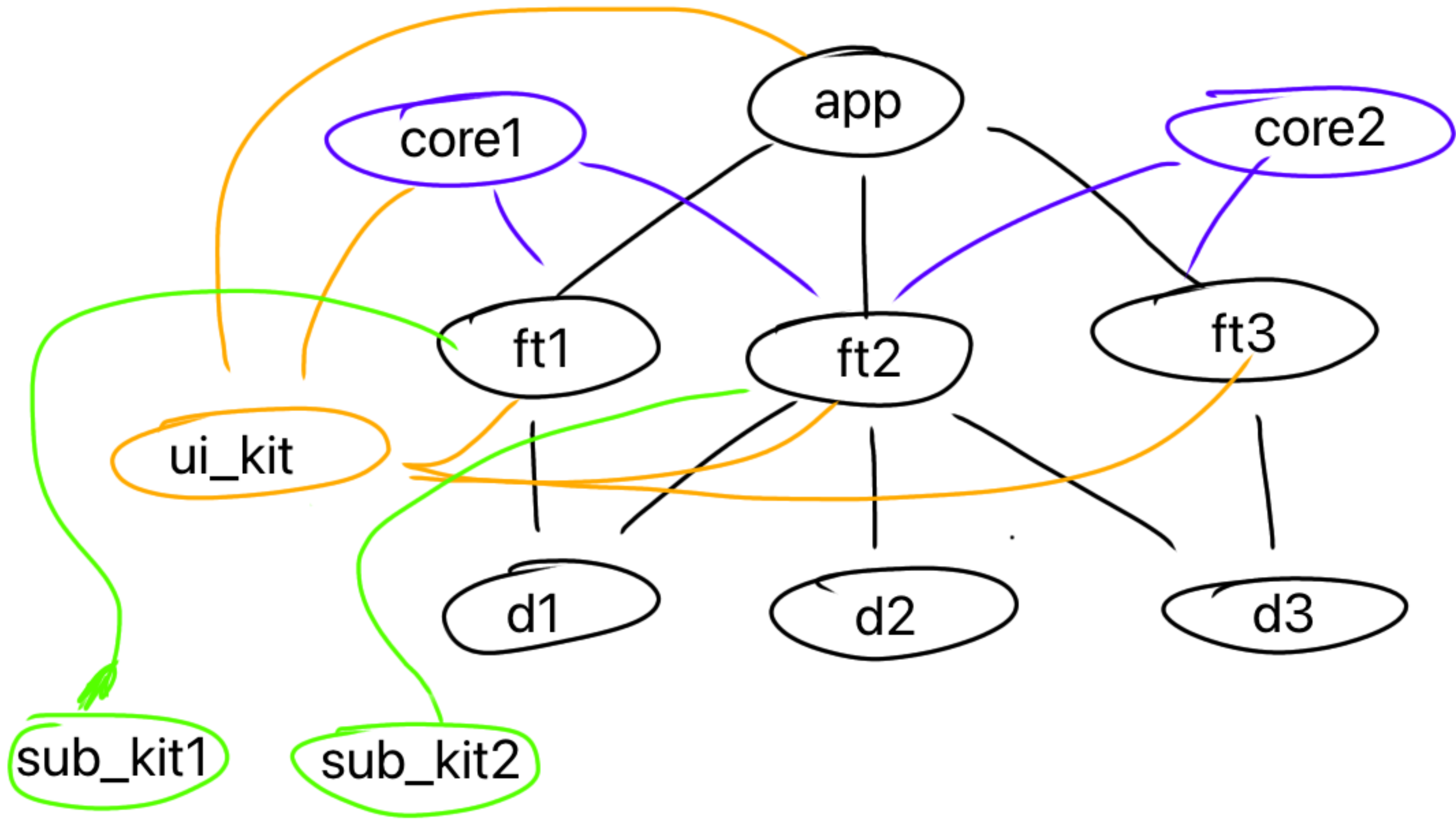
```
> :ui_kit:compileDebugKotlin  
> IDLE  
> IDLE  
> IDLE  
> IDLE  
> IDLE  
> IDLE  
> IDLE  
> IDLE
```

Изменение ABI →

Изменение API → **Ресбилд всех модулей вверху по цепочке**







Структура проекта — резюме

- Делайте более горизонтальную структуру проекта
- Избегайте бутылочных горлышек
- Избегайте избыточного создания модулей

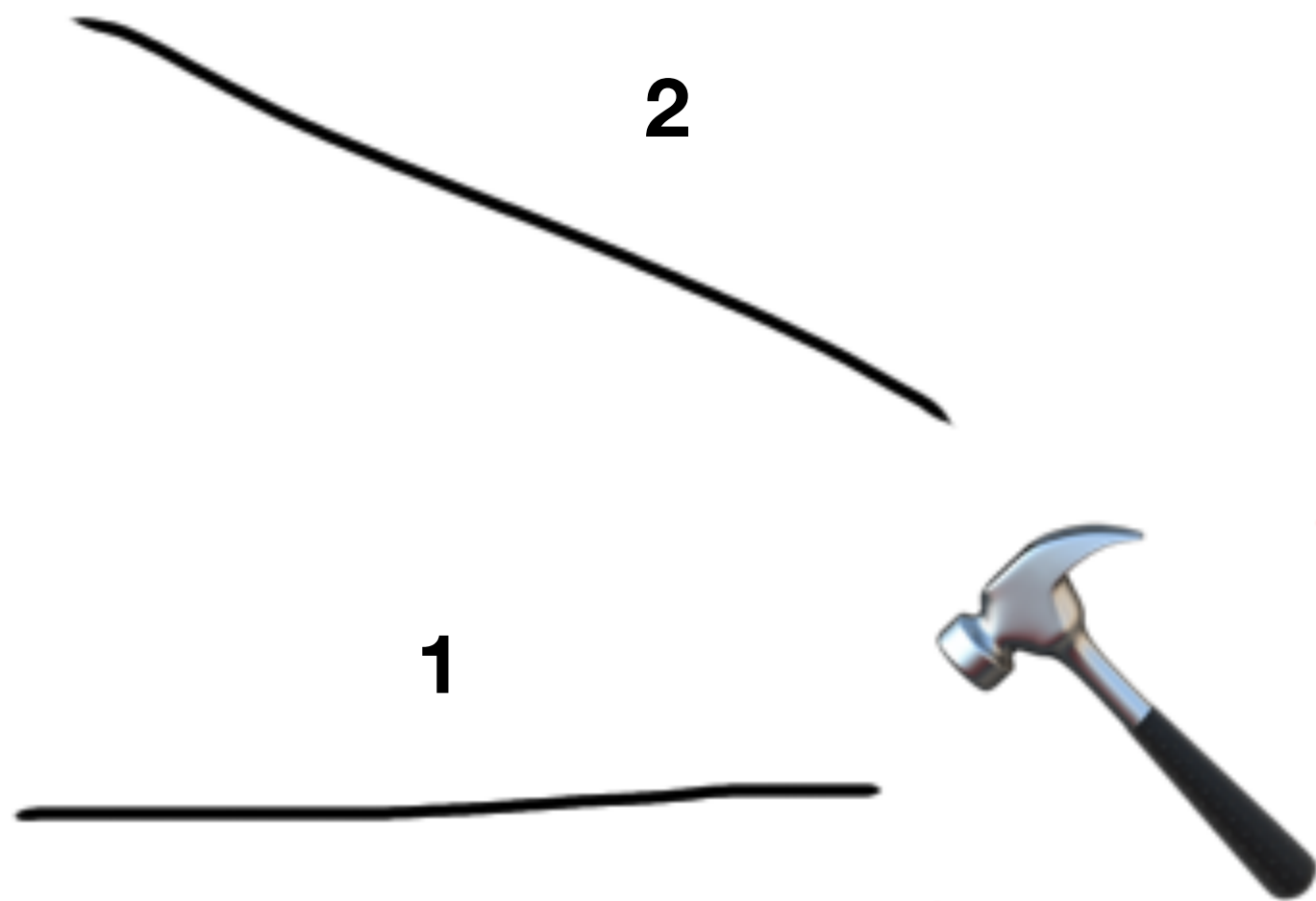


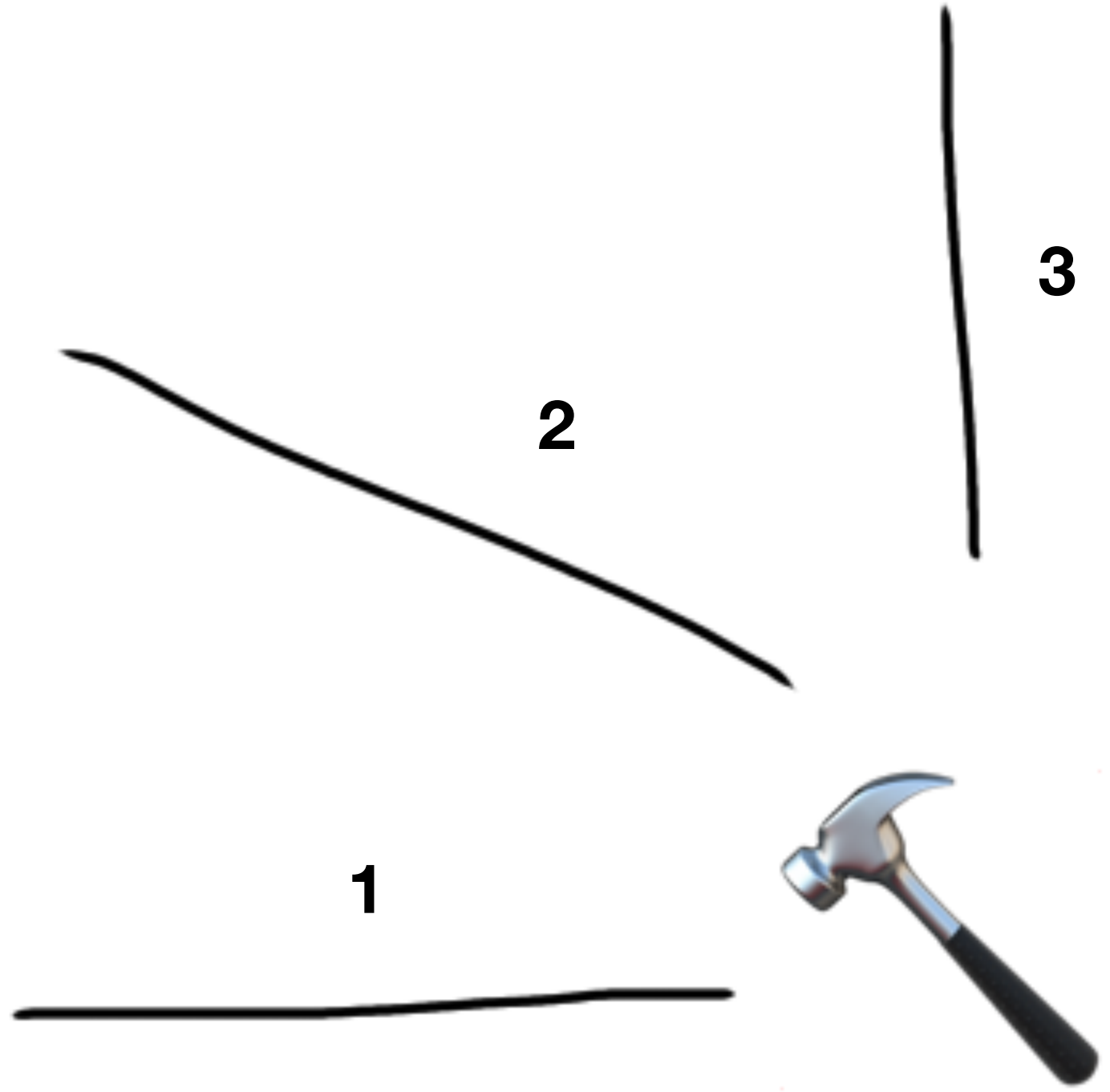
Build cache

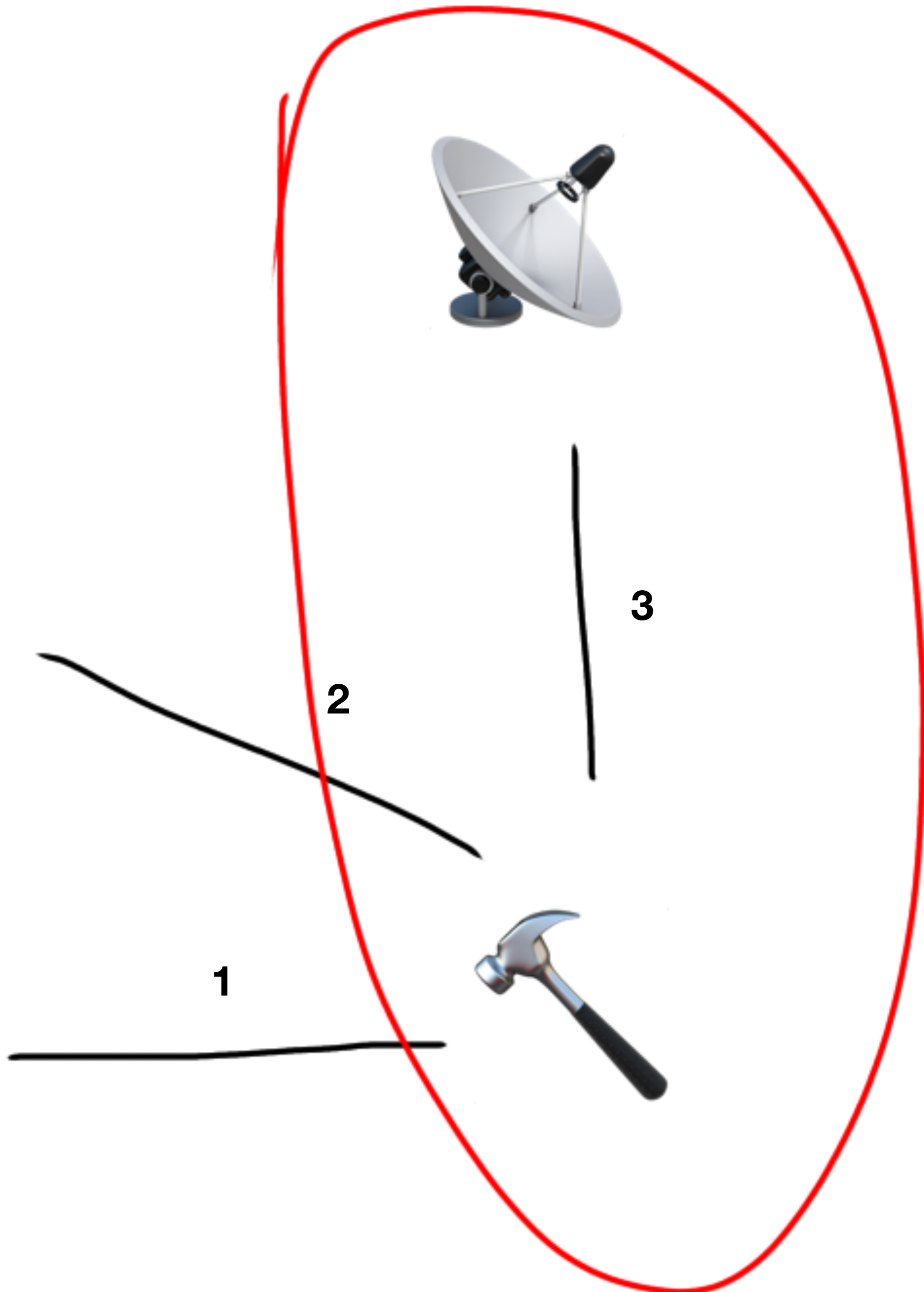




org.gradle.caching=true







```
buildCache {  
    def shouldEnablePush = isRunningOnCi && someOtherLogic  
    remote(HttpBuildCache) {  
        enabled = true  
        setPush(shouldEnablePush)  
        url = "someurl"  
    }  
}
```

- Используйте android-cache-fix Gradle плагин
- Делайте кастомные задачи кэшируемыми
- Унифицируйте окружение
 - JDK (например AdoptOpenJdk 11.0.10+9)
 - Ревизия android.jar
- Избавиться от динамического объявления зависимостей (**dependency:1.+**)



Ревизия android.jar

`$ANDROID_HOME/platforms/android-30/source.properties`

`Pkg.Revision`



Gradle Doctor

```
./gradlew :app:assembleDebug -PbenchmarkRemoteCache -PrerunSourceTasksForBenchmark -PrerunLargeOutputTasksForBenchmark
```

In order for a remote build cache to save you time, you would need an internet connection to your node of at least 0,77 MB/s.

In order for a remote build cache to save you time, you would need an internet connection to your node of **at least 0,77 MB/s**.



Про кэширование зависимостей билда



Обновите все что можно

Gradle, AGP, Kotlin, community plugins и т.д.

HO...



Будьте осторожны

./gradlew asDeb --scan

[Details](#)[Predecessors](#)[Successors](#)

Path	:common_kotlin:compileKotlin
Type	org.jetbrains.kotlin.gradle.tasks.KotlinCompile
Started after	24.288s
Duration >	10.754s

The task was not up-to-date because of the following reasons:

Output property 'destinationDirectory' file sources/common/common_kotlin/build/classes/kotlin/main has been removed.

Output property 'destinationDirectory' file sources/common/common_kotlin/build/classes/kotlin/main/META-INF has been removed.

Output property 'destinationDirectory' file sources/common/common_kotlin/build/classes/kotlin/main/META-INF/common_kotlin.kotlin_module has been removed.

Build cache result ▾	Miss (local and remote), Store (local)
Cache key	c634ecb6a1ec4fb83a699265636471d5
Cache artifact	14.4 KiB / 21 entries
Remote miss	0.211s
Pack	0.006s

[View task in console log](#)[Focus on task in timeline](#)

[Details](#)[Predecessors](#)[Successors](#)

Path	:common_kotlin:compileKotlin
Type	org.jetbrains.kotlin.gradle.tasks.KotlinCompile
Started after	24.288s
Duration >	10.754s

The task was not up-to-date because of the following reasons:

Output property 'destinationDirectory' file sources/common/common_kotlin/build/classes/kotlin/main has been removed.

Output property 'destinationDirectory' file sources/common/common_kotlin/build/classes/kotlin/main/META-INF has been removed.

Output property 'destinationDirectory' file sources/common/common_kotlin/build/classes/kotlin/main/META-INF/common_kotlin.kotlin_module has been removed.

Build cache result ▾

Cache key	Miss (local and remote), Store (local)
Cache artifact	c634ecb6a1ec4fb83a699265636471d5
Remote miss	14.4 KiB / 21 entries
Pack	0.211s
	0.006s

[View task in console log](#)[Focus on task in timeline](#)

[Details](#)[Predecessors](#)[Successors](#)

Path	:common_kotlin:compileKotlin
Type	org.jetbrains.kotlin.gradle.tasks.KotlinCompile
Started after	24.288s
Duration >	10.754s

The task was not up-to-date because of the following reasons:

Output property 'destinationDirectory' file sources/common/common_kotlin/build/classes/kotlin/main has been removed.

Output property 'destinationDirectory' file sources/common/common_kotlin/build/classes/kotlin/main/META-INF has been removed.

Output property 'destinationDirectory' file sources/common/common_kotlin/build/classes/kotlin/main/META-INF/common_kotlin.kotlin_module has been removed.

Build cache result ▾

Cache key	Miss local and remote Store (local)
Cache artifact	c634ecb6a1ec4fb83a699265636471d5
Remote miss	14.4 KiB / 21 entries
Pack	0.211s
	0.006s

[View task in console log](#)[Focus on task in timeline](#)



-Dorg.gradle.caching.debug=true

****LOCAL_GRADLE_HOME****/caches/modules-2/files-2.1/org.threeten/threetenbp/
1.5.0/736eaa38380d86f857b4ba455200f04fc611b4fe/threetenbp-1.5.0-no-
tzdb.jar=IGNORED / a935e5ad7ab5aa8f8021ab3af77f3c42

****CI_GRADLE_HOME****/caches/modules-2/files-2.1/org.threeten/threetenbp/
1.5.0/736eaa38380d86f857b4ba455200f04fc611b4fe/threetenbp-1.5.0-no-
tzdb.jar=IGNORED / 5fb6f0d802b1863971f84b057489751

****LOCAL_GRADLE_HOME****/caches/modules-2/files-2.1/org.threeten/threetenbp/1.5.0/736eaa38380d86f857b4ba455200f04fc611b4fe/threetenbp-1.5.0-no-tzdb.jar=IGNORED / a935e5ad7ab5aa8f8021ab3af77f3c42

****CI_GRADLE_HOME****/caches/modules-2/files-2.1/org.threeten/threetenbp/1.5.0/736eaa38380d86f857b4ba455200f04fc611b4fe/threetenbp-1.5.0-no-tzdb.jar=IGNORED / 5fb6f0d802b1863971f84b057489751

****LOCAL_GRADLE_HOME****/caches/modules-2/files-2.1/org.threeten/threetenbp/
1.5.0/736eaa38380d86f857b4ba455200f04fc611b4fe/threetenbp-1.5.0-no-
tzdb.jar=IGNORED / a935e5ad7ab5aa8f8021ab3af77f3c42

****CI_GRADLE_HOME****/caches/modules-2/files-2.1/org.threeten/threetenbp/
1.5.0/736eaa38380d86f857b4ba455200f04fc611b4fe/threetenbp-1.5.0-no-
tzdb.jar=IGNORED / 5fb6f0d802b1863971f84b057489751

****LOCAL_GRADLE_HOME****/caches/modules-2/files-2.1/org.threeten/threetenbp/1.5.0/736eaa38380d86f857b4ba455200f04fc611b4fe/threetenbp-1.5.0-no-tzdb.jar=IGNORED / a935e5ad7ab5aa8f8021ab3af77f3c42

****CI_GRADLE_HOME****/caches/modules-2/files-2.1/org.threeten/threetenbp/1.5.0/736eaa38380d86f857b4ba455200f04fc611b4fe/threetenbp-1.5.0-no-tzdb.jar=IGNORED / 5fb6f0d802b1863971f84b057489751

****LOCAL_GRADLE_HOME**/caches/modules-2/files-2.1/org.threeten/threetenbp/1.5.0/736eaa38380d86f857b4ba455200f04fc611b4fe/threetenbp-1.5.0-no-tzdb.jar=IGNORED / a935e5ad7ab5aa8f8021ab3af77f3c42**

****CI_GRADLE_HOME**/caches/modules-2/files-2.1/org.threeten/threetenbp/1.5.0/736eaa38380d86f857b4ba455200f04fc611b4fe/threetenbp-1.5.0-no-tzdb.jar=IGNORED / 5fb6f0d802b1863971f84b057489751**

****LOCAL_GRADLE_HOME****/caches/modules-2/files-2.1/org.threeten/threetenbp/1.5.0/736eaa38380d86f857b4ba455200f04fc611b4fe/threetenbp-1.5.0-no-tzdb.jar=IGNORED / **a935e5ad7ab5aa8f8021ab3af77f3c42**

****CI_GRADLE_HOME****/caches/modules-2/files-2.1/org.threeten/threetenbp/1.5.0/736eaa38380d86f857b4ba455200f04fc611b4fe/threetenbp-1.5.0-no-tzdb.jar=IGNORED / **5fb6f0d802b1863971f84b057489751**



Gradle Community Slack

- Gradle сделали баг в 6.8 который не замечали до 6.8.2
- Из фикса этого бага они сделали отдельный релиз 6.8.3
- Всегда мониторьте ваши билды после апдейта зависимостей билдскриптов
- Пользуйтесь Gradle Community Slack

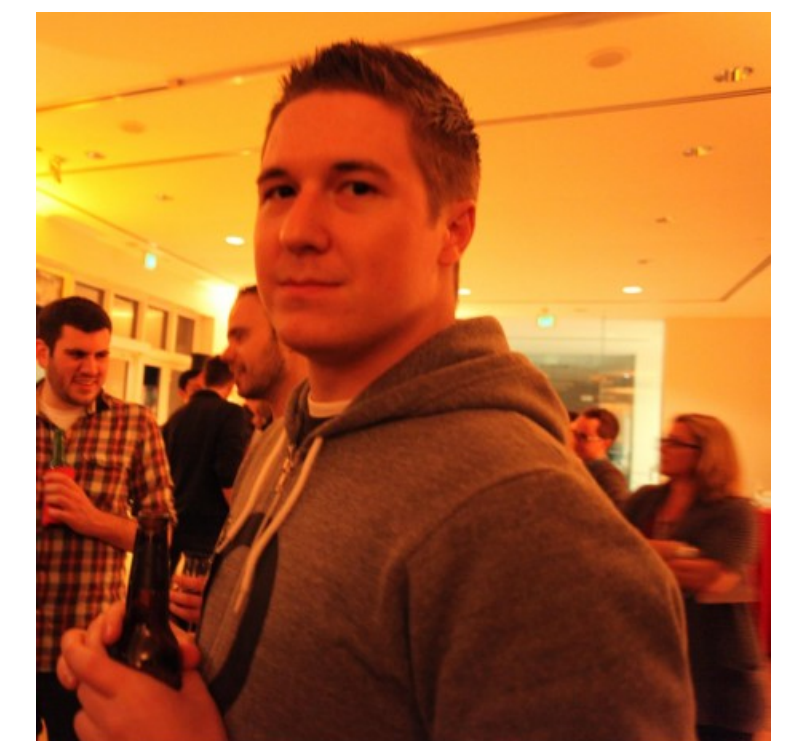
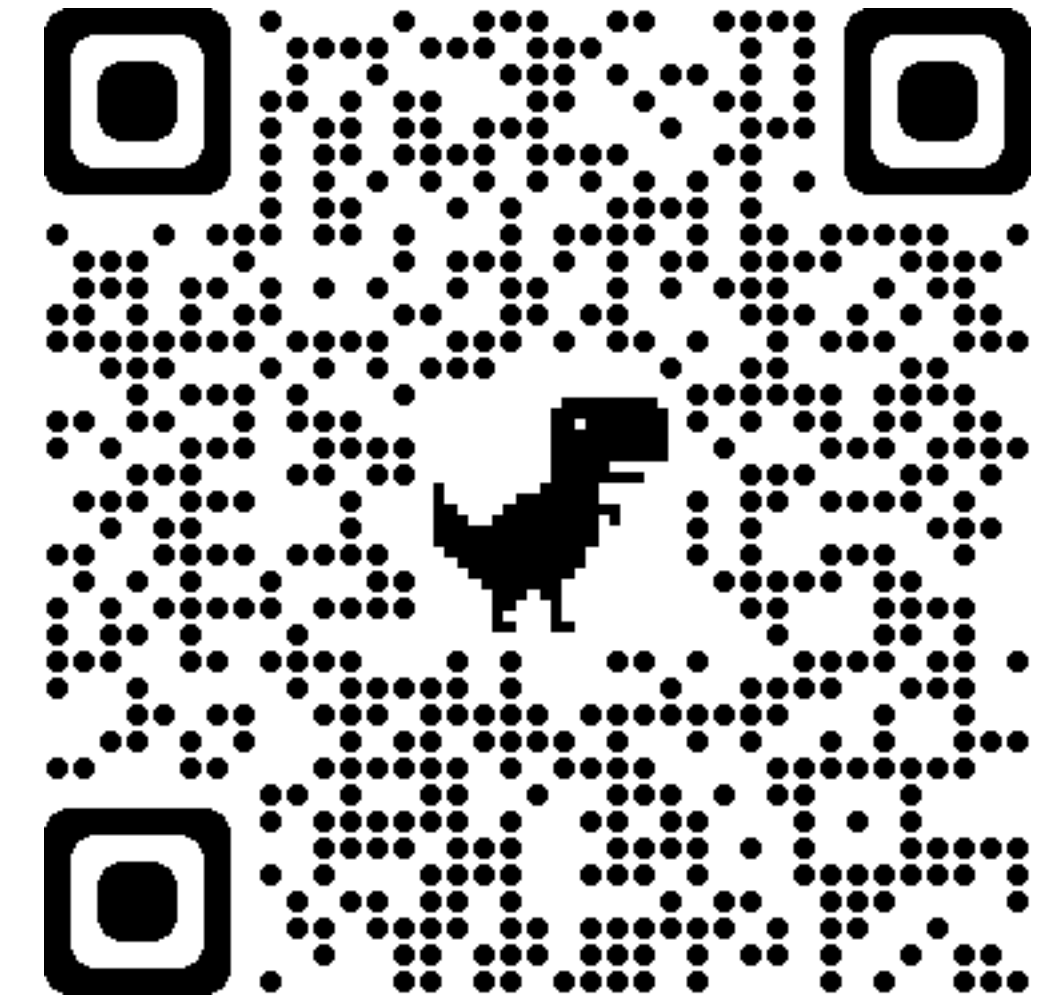
Annotation processing

Dagger 2

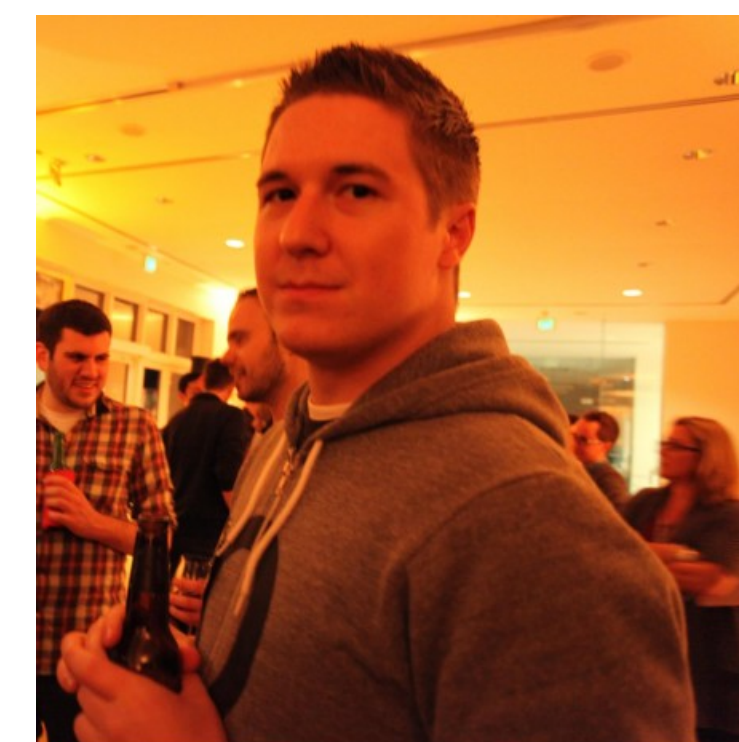


Dagger Reflect

- Библиотека от Jake Wharton
- Частичная рефлексия
- Полная рефлексия



- Библиотека от Jake Wharton
- Частичная рефлексия ✓
- Полная рефлексия



- Используйте `@Component.Builder` / `@Component.Factory` для создания компонентов
- Убедитесь что ваши аннотации `@Qualifier`, `@MapKey` и другие используют runtime retention

```
if (properties.containsKey('android.injected.invoked.from.ide')) {
    debugAnnotationProcessor 'com.jakewharton.dagger:dagger-reflect-compiler:0.3.0'
    debugApi 'com.jakewharton.dagger:dagger-reflect:0.3.0' // or debugImplementation
} else {
    debugAnnotationProcessor "com.google.dagger:dagger-compiler:$daggerVersion"
}
releaseAnnotationProcessor "com.google.dagger:dagger-compiler:$daggerVersion"
api "com.google.dagger:dagger:$daggerVersion" // or implementation
```

Есть способ лучше?

Delect

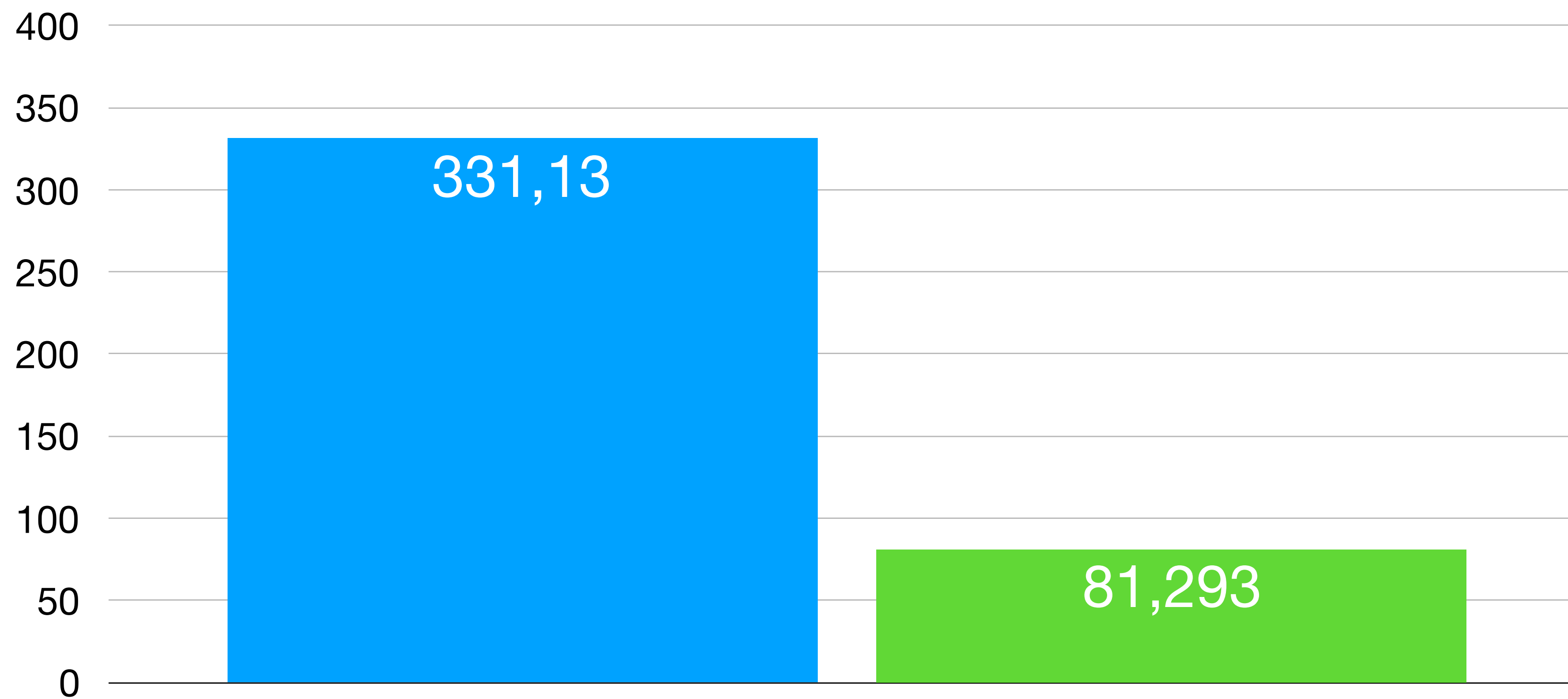
- Gradle плагин
- Использует `dependency substitution API`
- Применяется 1 раз

```
buildscript {  
    dependencies {  
        classpath("com.soundcloud.delect:delect-plugin:0.3.0")  
    }  
}  
  
apply plugin: "com.soundcloud.delect"
```

```
strategy.dependencySubstitution { substitution →  
    substitution  
        .substitute(substitution.module("$daggerGroupId:dagger-compiler"))  
        .with(substitution.module("$reflectDaggerGroupId:dagger-reflect-compiler:${extension.daggerReflectVersion}"))  
}
```


Результаты

■ Без Dagger Reflect ■ С Dagger Reflect
Время AP (сек)



**Уменьшение времени AP
на ~75%**

Минусы?

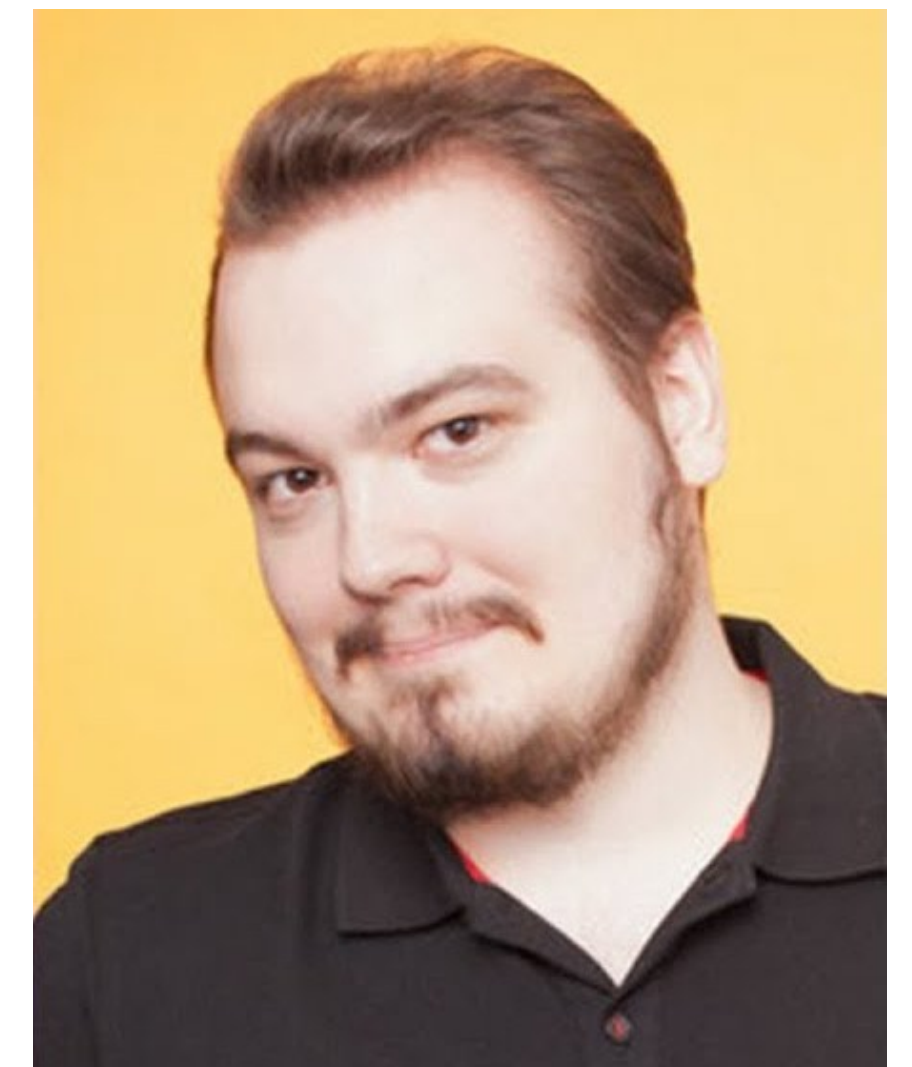
- Отсутствие compile time валидации DI графа
- Импакт на производительность

**Не забудьте прогреть remote build cache билдом с включенным
Dagger Reflect**



NAPT

- Плагин Java компилятора
- Позволяет напрямую использовать Java APT с Kotlin классами
- Оригинальная идея — Владимир Тагаков



Java APT

KAPT

NAPT


 Java classes

 Java stubs

 Kotlin classes

 Java APT

 Java APT

 Kotlin classes

Java стабы

- Java классы на основе Kotlin исходников
- Содержат публичные и **приватные** методы/поля
- Генерируются даже для классов без аннотаций

```
javac [ options ] [ sourcefiles ] [ classes ] [ @argfiles ]
```

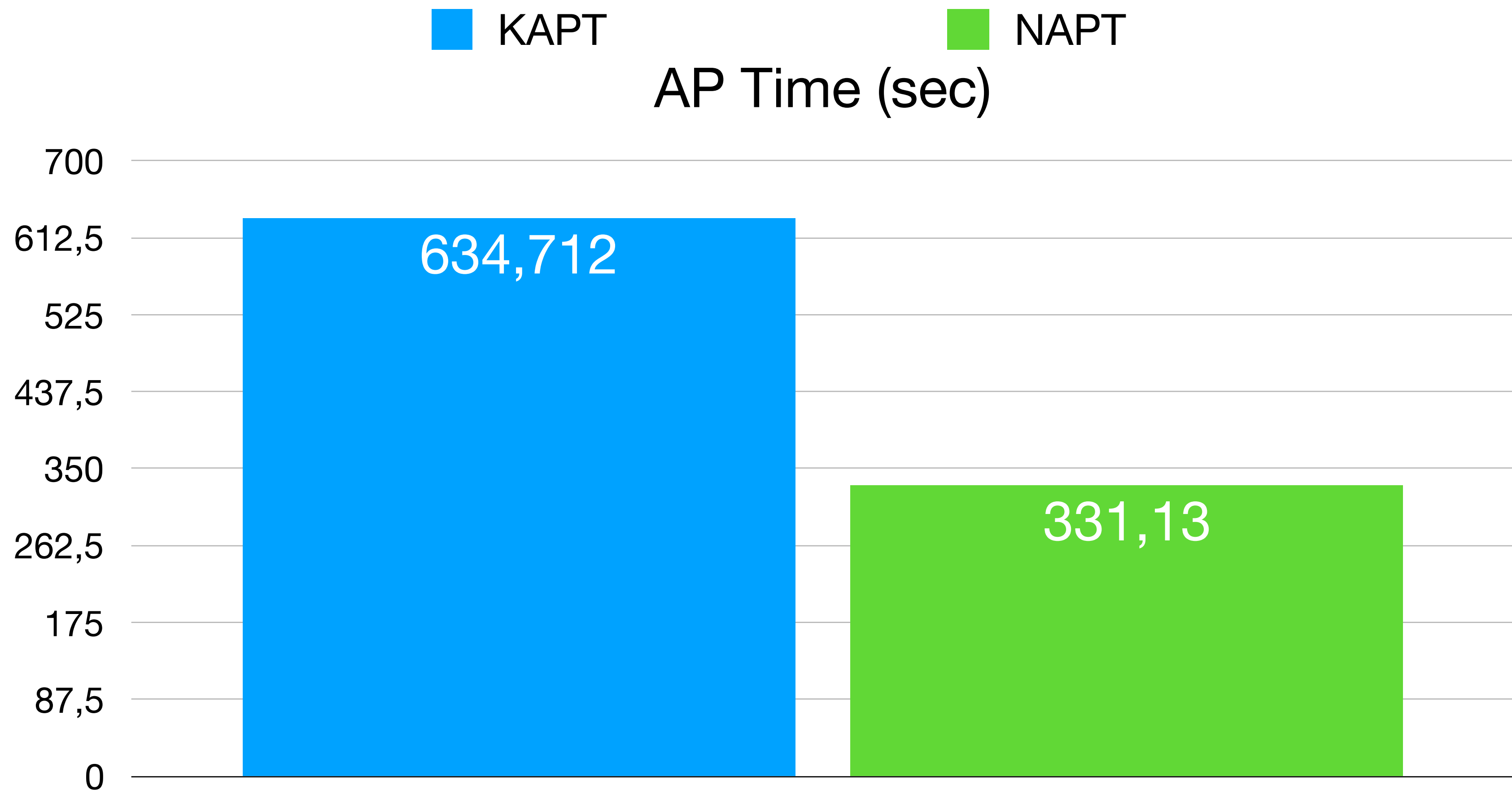
```
javac [ options ] [ sourcefiles ] [ classes ] [ @argfiles ]
```

classes

One or more classes to be processed for annotations (such as MyPackage.MyClass)

```
tasks.withType(JavaCompile::class.java).configureEach { task: JavaCompile →  
    task.doFirst {  
        task.options.annotationProcessorPath = task.options.annotationProcessorPath?.plus(path-to-napt-jar)  
        task.options.compilerArgs = task.options.compilerArgs + listOf("-Xplugin:Napt")  
    }  
}
```

```
tasks.withType(JavaCompile::class.java).configureEach { task: JavaCompile →  
    task.doFirst {  
        task.options.annotationProcessorPath = task.options.annotationProcessorPath?.plus(path-to-napt-jar)  
        task.options.compilerArgs = task.options.compilerArgs + listOf("-Xplugin:Napt")  
    }  
}
```



~50% уменьшение времени AP

В чем подвох?

```
public class DaggerRootComponentBridge {  
    public static RootComponent.Builder builder() {  
        return DaggerRootComponent.builder();  
    }  
}
```

***Проверено только для Dagger 2**

Бонус: NAPT vs KSP

Kotlin Symbol Processing

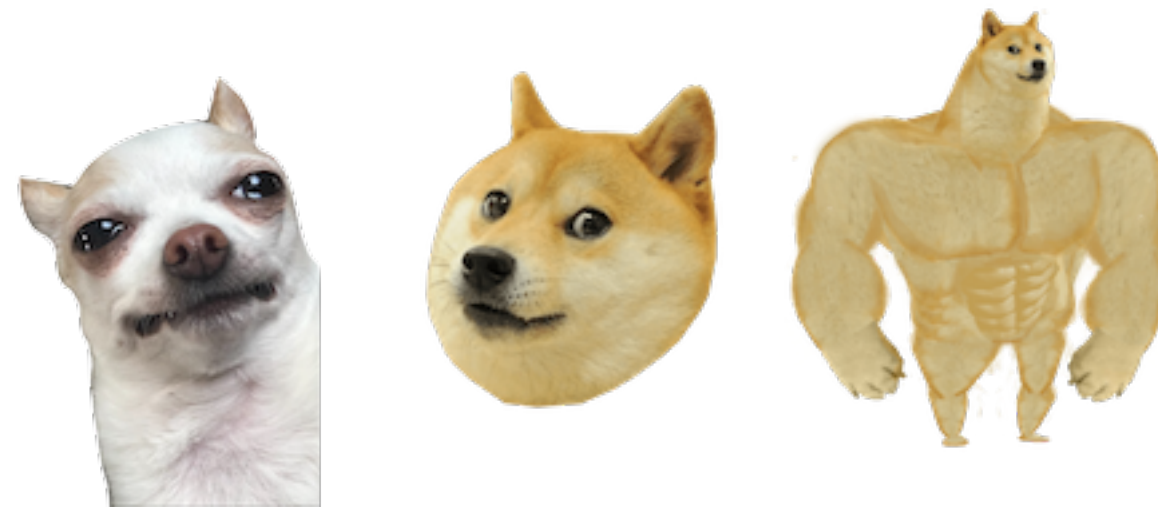
KSP

- Новое API для разработки легковесных плагинов Kotlin компилятора
- Аннотейшн процессоры с поддержкой KSP до 2х раз быстрее чем с KAPT
- Нет привязки к JVM

KSP

- Новое API для разработки легковесных плагинов Kotlin компилятора
- Аннотейшн процессоры с поддержкой KSP до 2х раз быстрее чем с KAPT
- **Нет привязки к JVM**

Library	Status
Room	<u>Experimentally supported</u>
Moshi	<u>Experimentally supported</u>
Auto Factory	Not yet supported
Dagger	Not yet supported
Hilt	Not yet supported
Glide	Not yet supported



buildSrc

Контент buildSrc

- Объявления зависимостей
- Precompiled script plugins
- Утилитные задачи

Любое изменение (ABI, non-ABI) →

Любое изменение (ABI, non-ABI) →



Решение?

Композитный билд

I'll put a build inside your build



To build it while you're building

- Несвязанная логика, разнесенная по разным модулям
- Предсказуемая инкрементальность билдов
- Удобство (с Gradle 6.8)
 - `./gradlew :plugins:napt:java-11:assemble`
- Супер удобство (с Gradle 7.0)
 - Convention плагины для Settings

- Несвязанная логика, разнесенная по разным модулям
- **Предсказуемая инкрементальность билдов**
- Удобство (с Gradle 6.8)
 - `./gradlew :plugins:napt:java-11:assemble`
- Супер удобство (с Gradle 7.0)
 - Convention плагины для Settings

- Если ваш buildSrc простой и не изменяется постоянно
- Если вы используете Gradle 6.8+ (были проведены некоторые оптимизации по compilation avoidance)
- Если у вас нет buildSrc в проекте

**Возможно вам не нужен
КОМПОЗИТНЫЙ билд**

**Возможно вам не нужен
КОМПОЗИТНЫЙ билд**



Отключение фиич AGP

- Уменьшает количество тасок (регистрируемых, выполняемых) на ~15%
- На ~15 секунд сократилось время чистого билда (~120 Android модулей)


```
android {  
    buildFeatures {  
        buildConfig = true  
        aidl = true  
        renderScript = true  
        shaders = true  
        resValues = true  
    }  
}
```



Jetifier

SUPPORT - V7...





ANDROID X...



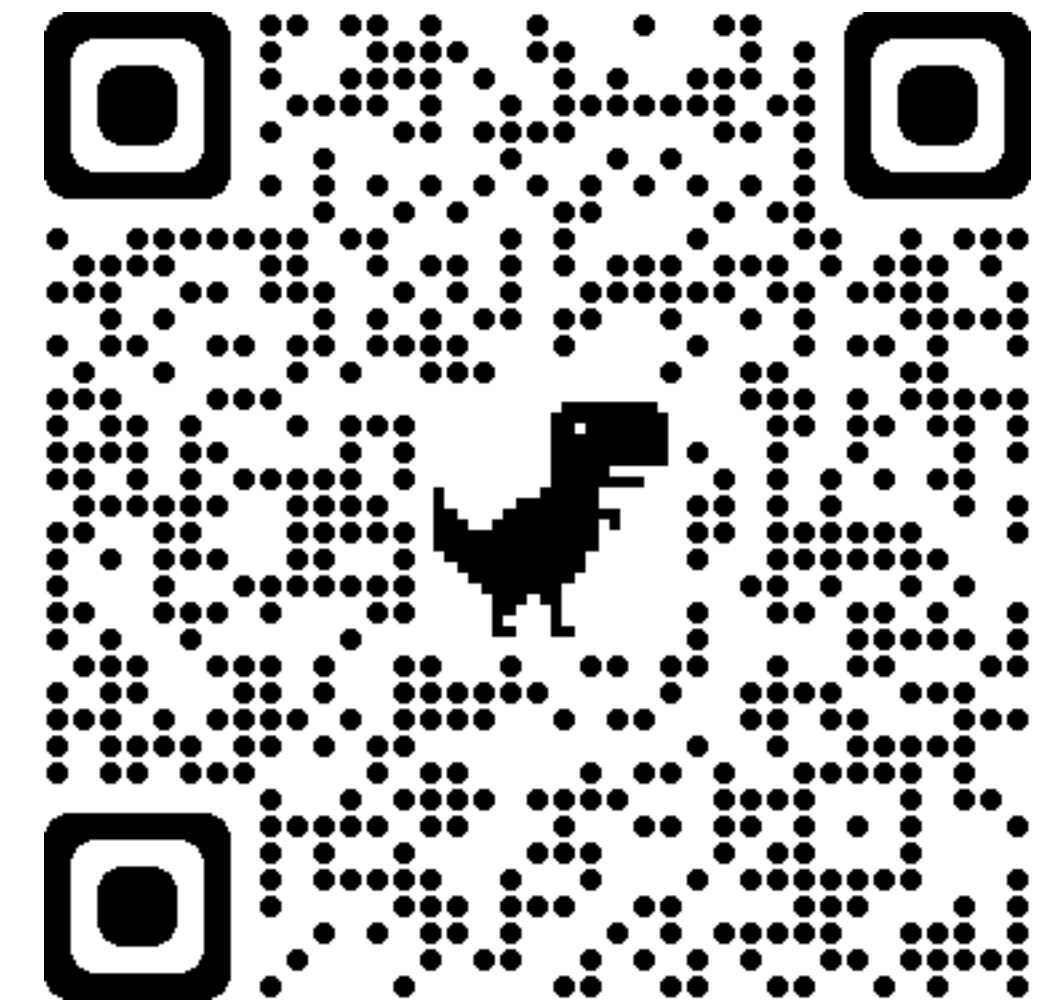
Работает не быстро...

`android.enableJetifier=true`

`android.enableJetifier=false`

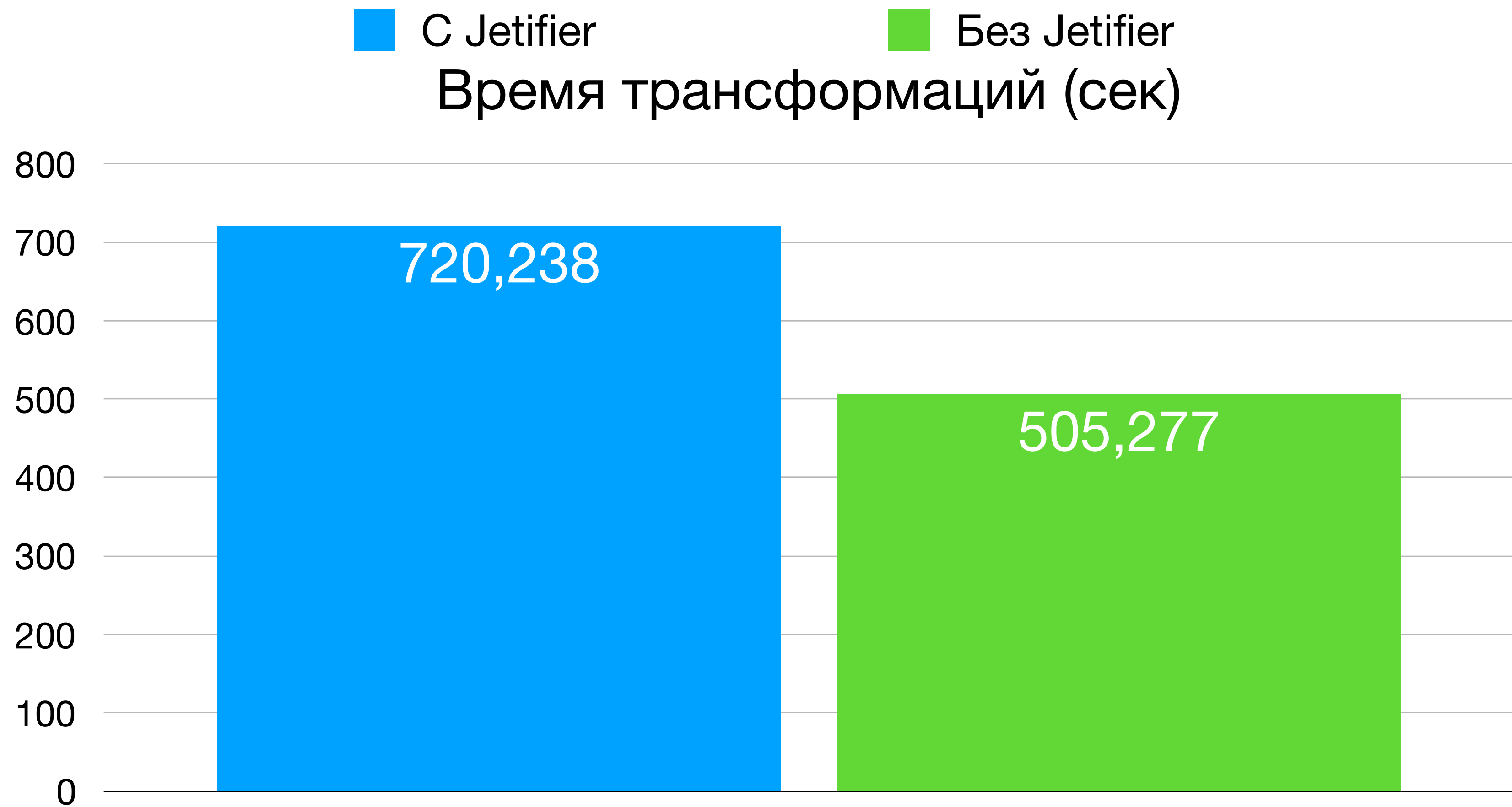


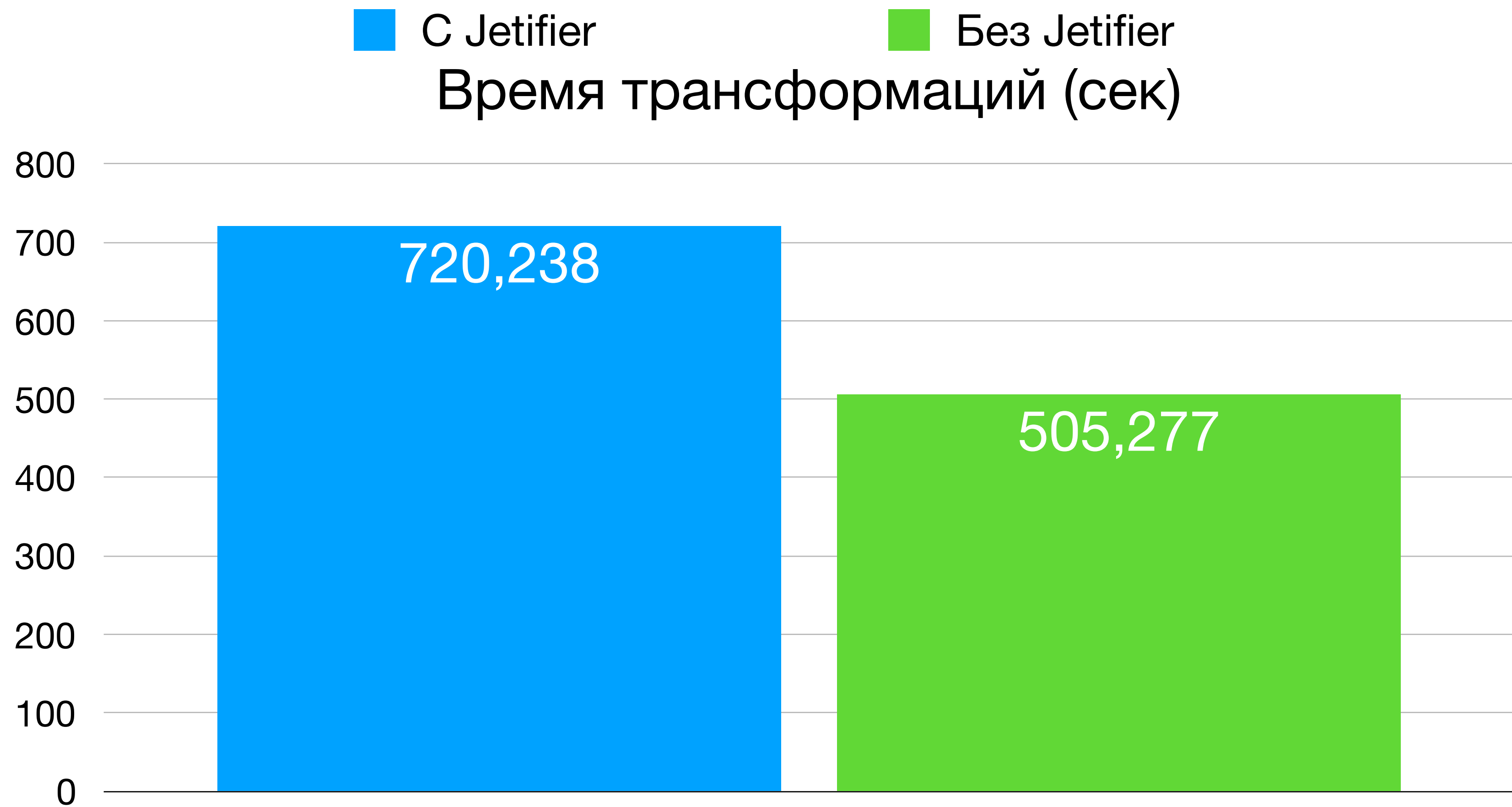
- CanIDropJetifier Gradle плагин
- Выявить библиотеки, требующие джетифаинга
- Джетифайнуть руками
 - jetifier-standalone
 - Используйте маппинги
- Загрузите в nexus/artifactory/mavenLocal
- **Проверьте fatJar зависимости**



- 6 артефактов
- ~48 mb

Результаты





**~30% всего времени трансформации
занимал JetifyTransform**

Now then...

Основные моменты (Апрель 2021)

- KLOC: ~**380k** (+ 150%)
- Gradle модули: ~**290** (+ 190%)
- Композитный билд (**buildSrc**)
- Чистый билд: ~**4.3 min** (- ~40%)
- Инкрементальный билд: ~**1.2 min** (- 60%)

P.S.

Неназванное

aar2jar+androidJar



- Перевод android library модулей в pure kotlin модули
- Оригинальная идея — Степан Гончаров



Flavour explosion

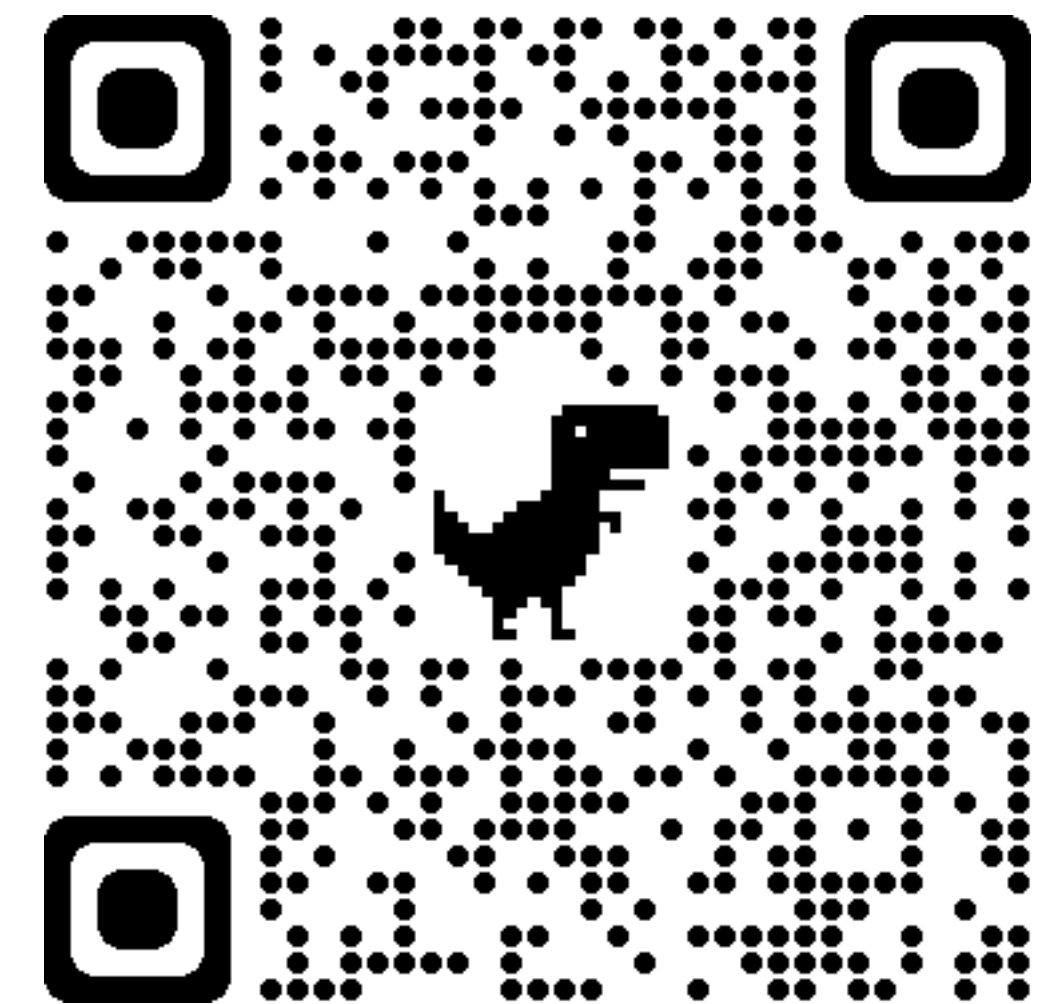
```
android {  
    variantFilter {  
        if (name == "release") ignore = true  
    }  
}
```

```
android {  
    buildTypes {  
        getByName("release") {  
            setMatchingFallbacks("debug")  
        }  
    }  
}
```

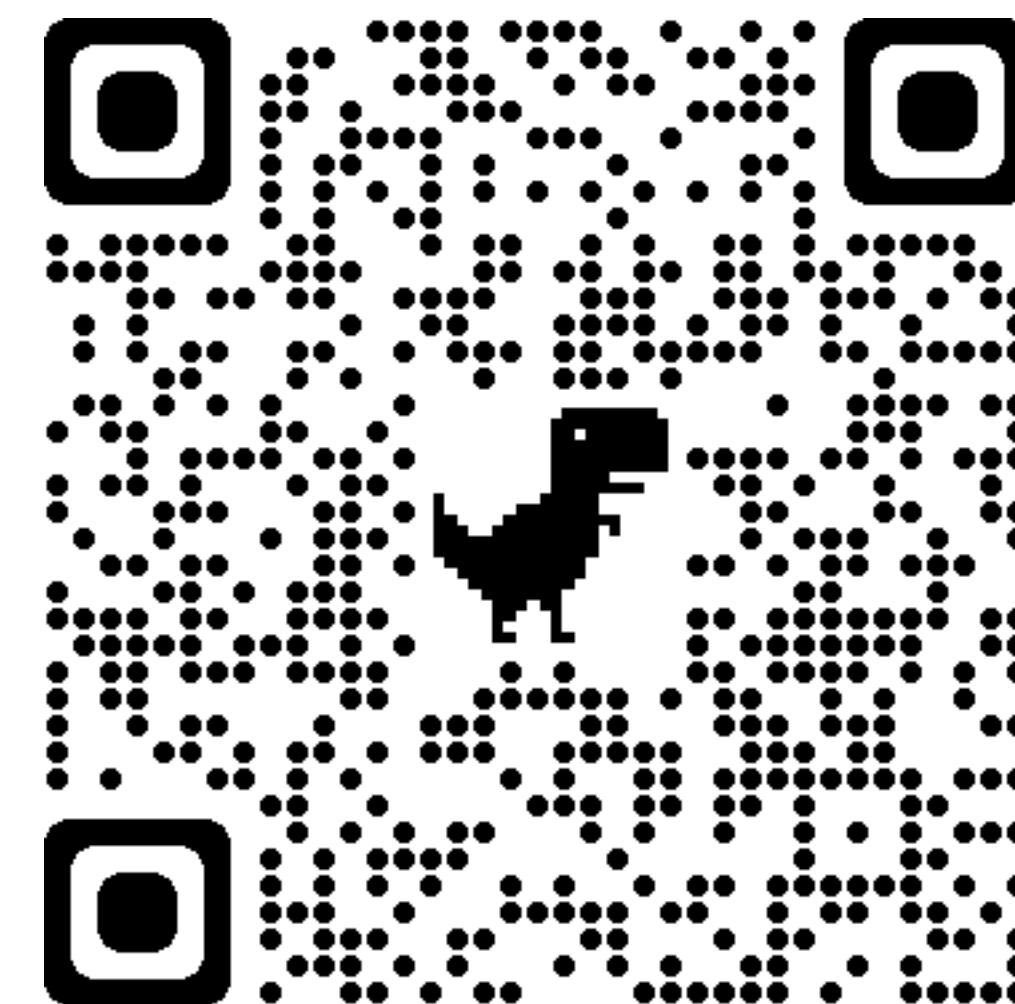
- aar2jar + androidJar
- Flavour explosion
- engBuild
- Configuration cache
- Жонглирование jvmargs
- Gradle profiler
- Remote builds
- ...

One more thing

НАРТ можно использовать



Вопросы?



[sergey_lapin_](#)



[elbermendax](#)