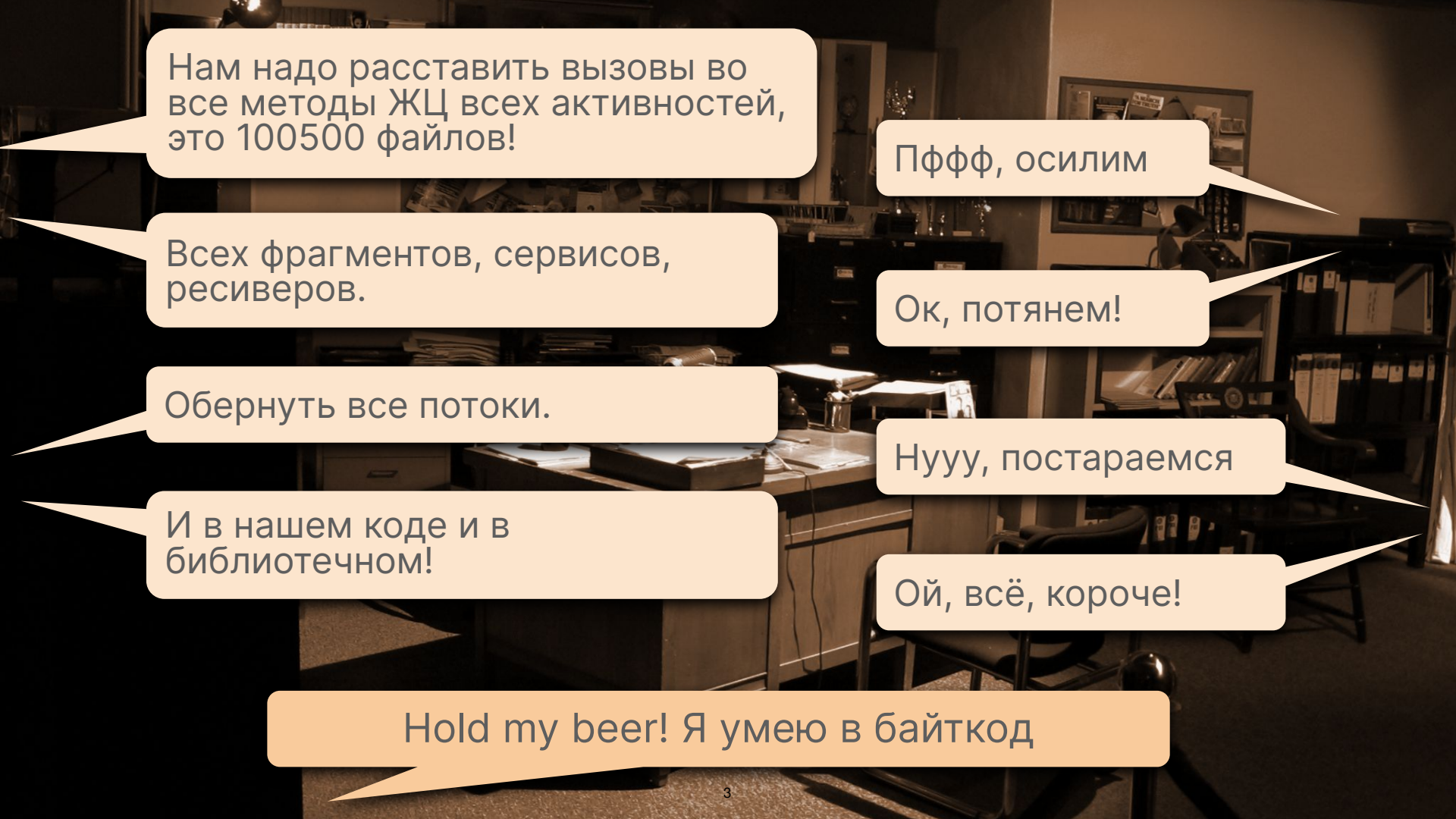
OK.TECH

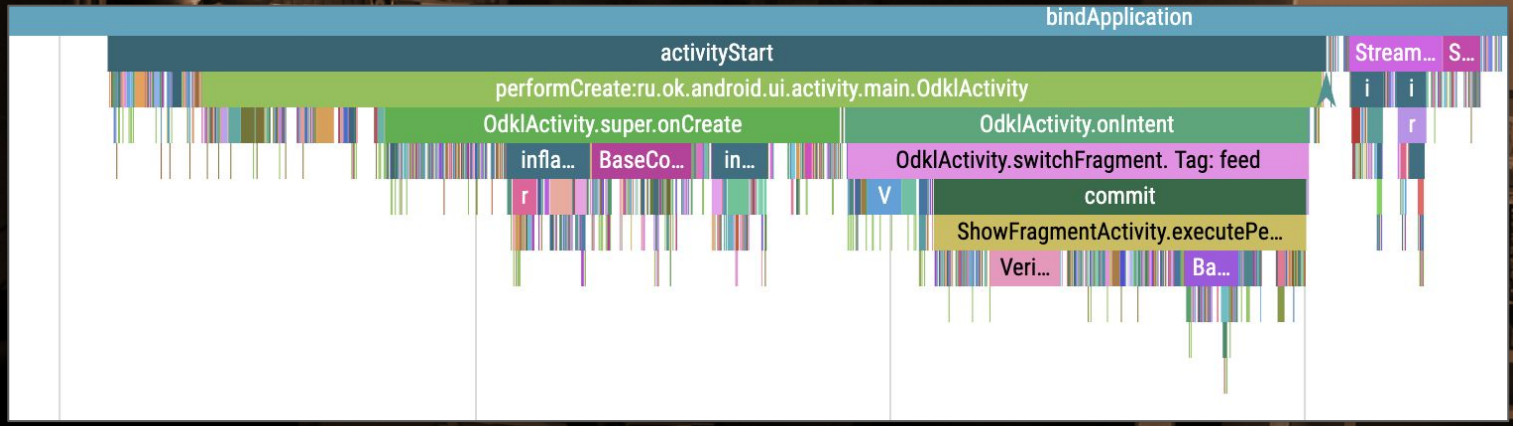
ByteWeaver

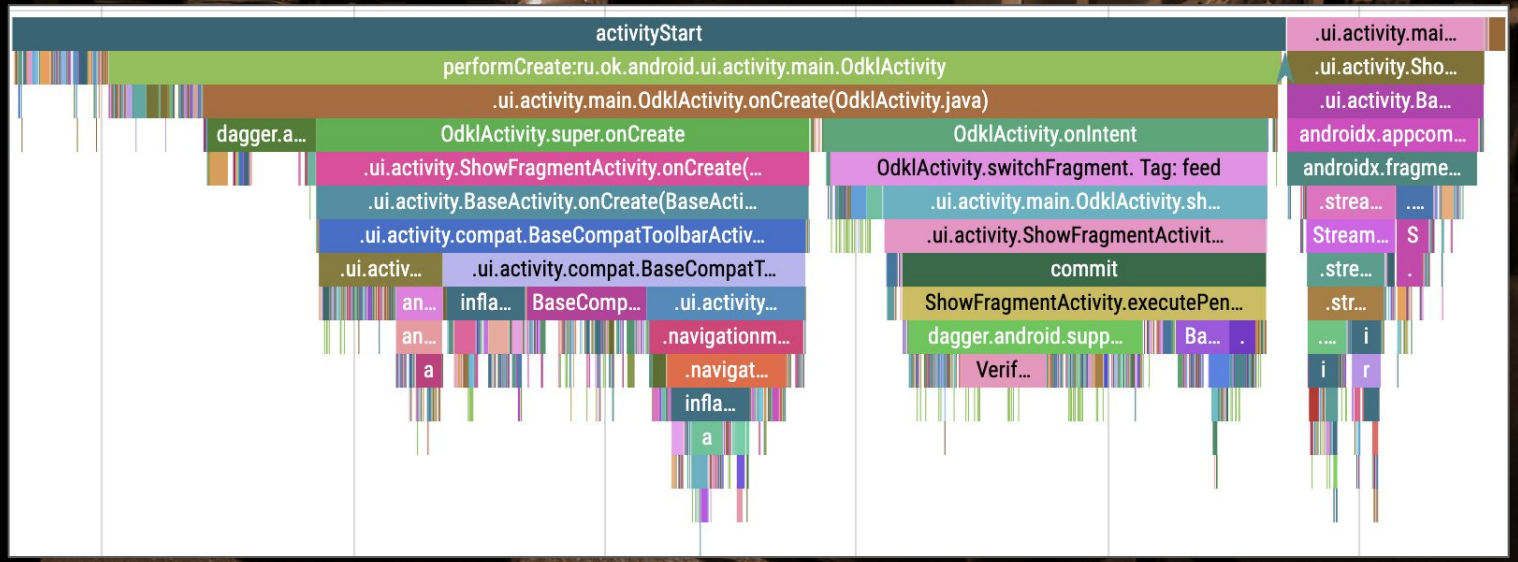
Инструментирование байткода во имя великого блага

Александр Асанов, Android разработчик в OK, Tracer, ByteWeaver


Итак, мы умеем собирать данные о вызовах методов через systrace на устройствах пользователя *в продакшене*!
YAY! 🎉
Ничоси 😲
Это же круто!
Да, но есть нюанс. Данных в трейсах мало, потому что мало где расставлены вызовы **Trace.beginSection** и **Trace.endSection**
А! Ну всё, капец!
Да неее, расставим!
А вот и не расставим!
XX18 год, офис Одноклассников


Нам надо расставить вызовы во все методы ЖЦ всех активностей, это 100500 файлов!
Пффф, осилим
Всех фрагментов, сервисов, ресиверов.
Ок, потянем!
Обернуть все потоки.
Нууу, постараемся
И в нашем коде и в библиотечном!
Ой, всё, короче!
Hold my beer! Я умею в байткод


bindApplication
activityStart
Stream…
S…
performCreate:ru.ok.android.ui.activity.main.OdklActivity
i
i
OdklActivity.super.onCreate
OdklActivity.onIntent
r
infla…
BaseCo…
in…
OdklActivity.switchFragment. Tag: feed
r
V
commit
ShowFragmentActivity.executePe…
Veri…
Ba…


activityStart
.ui.activity.mai…
performCreate:ru.ok.android.ui.activity.main.OdklActivity
.ui.activity.Sho…
.ui.activity.main.OdklActivity.onCreate(OdklActivity.java)
.ui.activity.Ba…
dagger.a…
OdklActivity.super.onCreate
OdklActivity.onIntent
androidx.appcom…
.ui.activity.ShowFragmentActivity.onCreate(…
OdklActivity.switchFragment. Tag: feed
androidx.fragme…
.ui.activity.BaseActivity.onCreate(BaseActi…
.ui.activity.main.OdklActivity.sh…
.strea…
.ui.activity.compat.BaseCompatToolbarActiv…
.ui.activity.ShowFragmentActivit…
Stream…
S
.ui.activ…
.ui.activity.compat.BaseCompatT…
commit
.stre…
an…
infla…
BaseComp…
.ui.activity…
ShowFragmentActivity.executePen…
.str…
an…
.navigationm…
dagger.android.supp…
Ba…
i
a
.navigat…
Verif…
i
r
infla…
a


Абалдеть! А я хочу поймать все тосты в тестах...
А я хочу залогировать все показы всех уведомлений, а то пользователи жалуются...
А у меня тут крэш непонятный. Памагити!..

OK.TECH

Так, стоп!

Давай-ка по порядку.

План

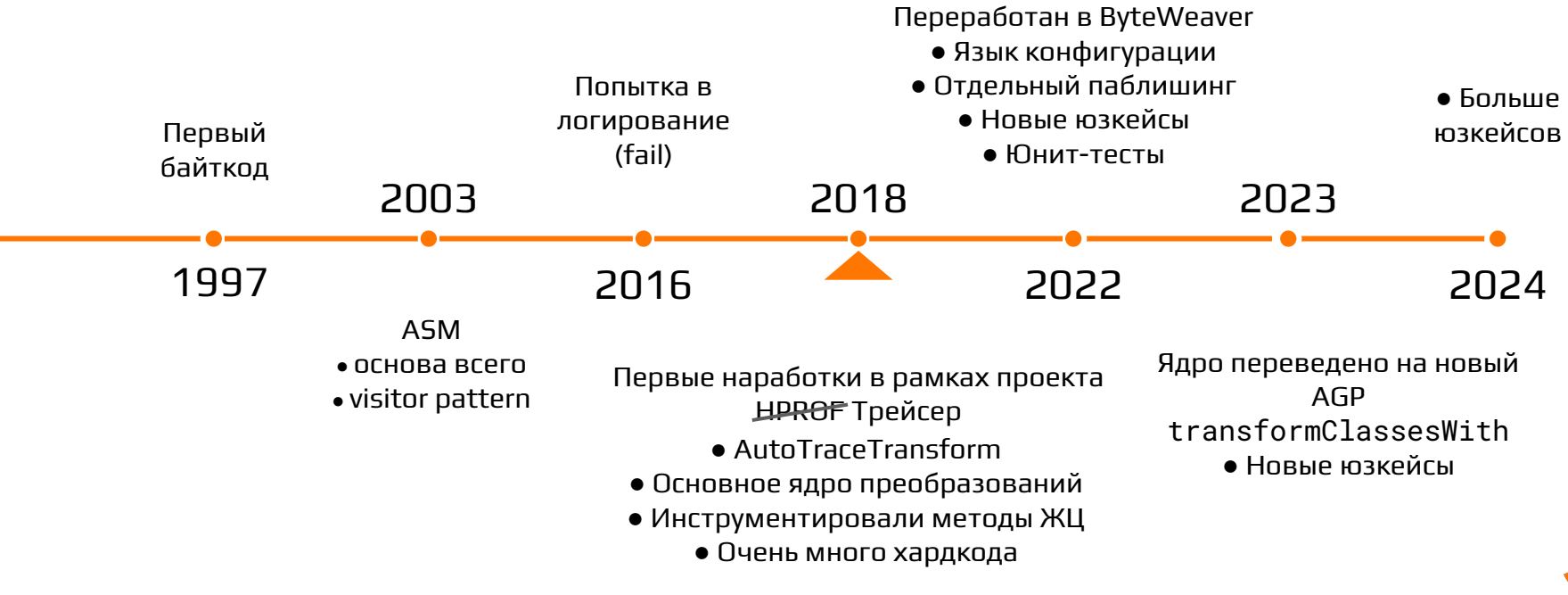
- История ByteWeaver
- Что такое байткод
- Когда мы будем его править
- Каким образом мы будем это делать
- Истории из жизни
- Выводы
- Планы на будущее

OK.TECH

История

Откуда есть пошел ByteWeaver

История ByteWeaver

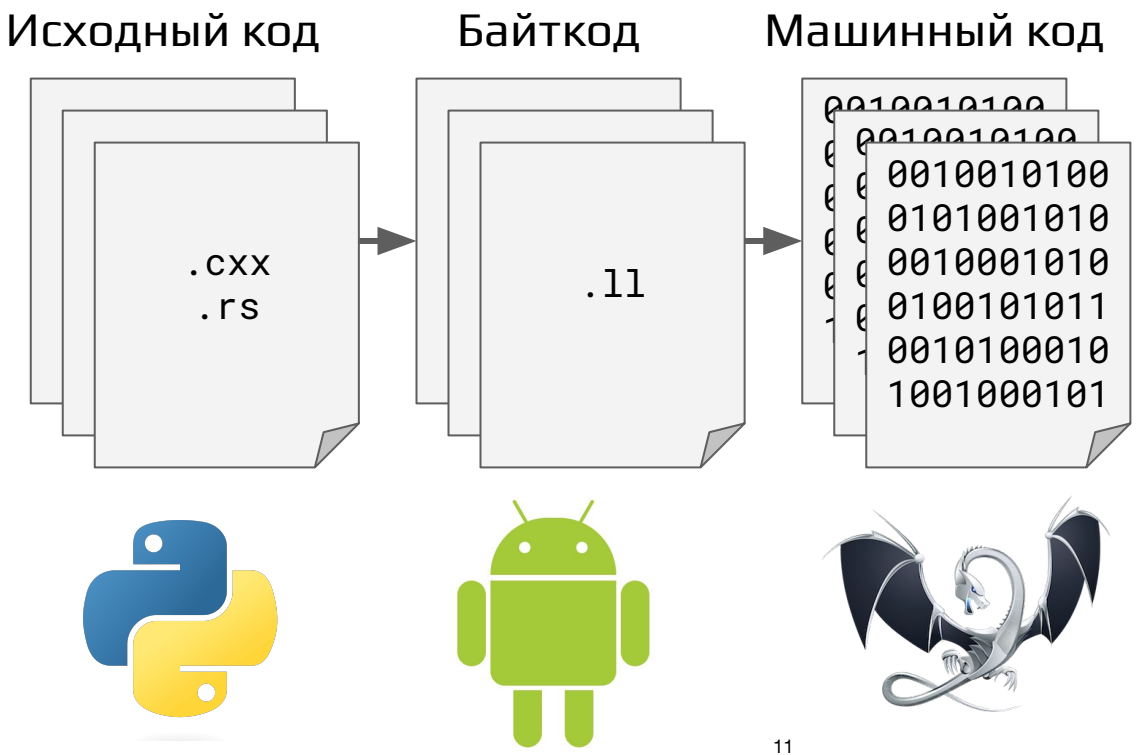


OK.TECH

Что такое байткод

немного теории

Код → Код → Код



- Интерпретация
- JIT
- AOT
- Компиляция

Пример байткода

```
class Example {
    fun execute(runnable: Runnable): Int {
        try {
            println("Going to run")
            runnable.run()
        } catch (ex: Throwable) {
            println("What a Terrible Failure")
        }
        return 0
    }
}
```

```
public final int execute(java.lang.Runnable);
  Code:
     0: aload_1
     1: ldc           #17                 // String runnable
     3: invokestatic  #23                 // Method kotlin/jvm/internal/Intrinsics.checkNotNullParameter:
                                          //                      (Ljava/lang/Object;Ljava/lang/String;)V
     6: nop
     7: ldc           #25                 // String Going to run
     9: getstatic     #31                 // Field java/lang/System.out:Ljava/io/PrintStream;
    12: swap
    13: invokevirtual #37                 // Method java/io/PrintStream.println:(Ljava/lang/Object;)V
    16: aload_1
    17: invokeinterface #42,  1           // InterfaceMethod java/lang/Runnable.run:()V
    22: goto          35
    25: astore_2
    26: ldc           #44                 // String What a Terrible Failure
    28: getstatic     #31                 // Field java/lang/System.out:Ljava/io/PrintStream;
    31: swap
    32: invokevirtual #37                 // Method java/io/PrintStream.println:(Ljava/lang/Object;)V
    35: iconst_0
    36: ireturn
  Exception table:
     from    to  target type
         6    22    25   Class java/lang/Throwable
```

SPOILER ALERT!!!

ByteWeaver патчит начало/конец функции и вызовы функций

Пример байткода (Dalvik)

```
class Example {
    fun execute(runnable: Runnable): Int {
        try {
            println("Going to run")
            runnable.run()
        } catch (ex: Throwable) {
            println("What a Terrible Failure")
        }
        return 0
    }
}
```

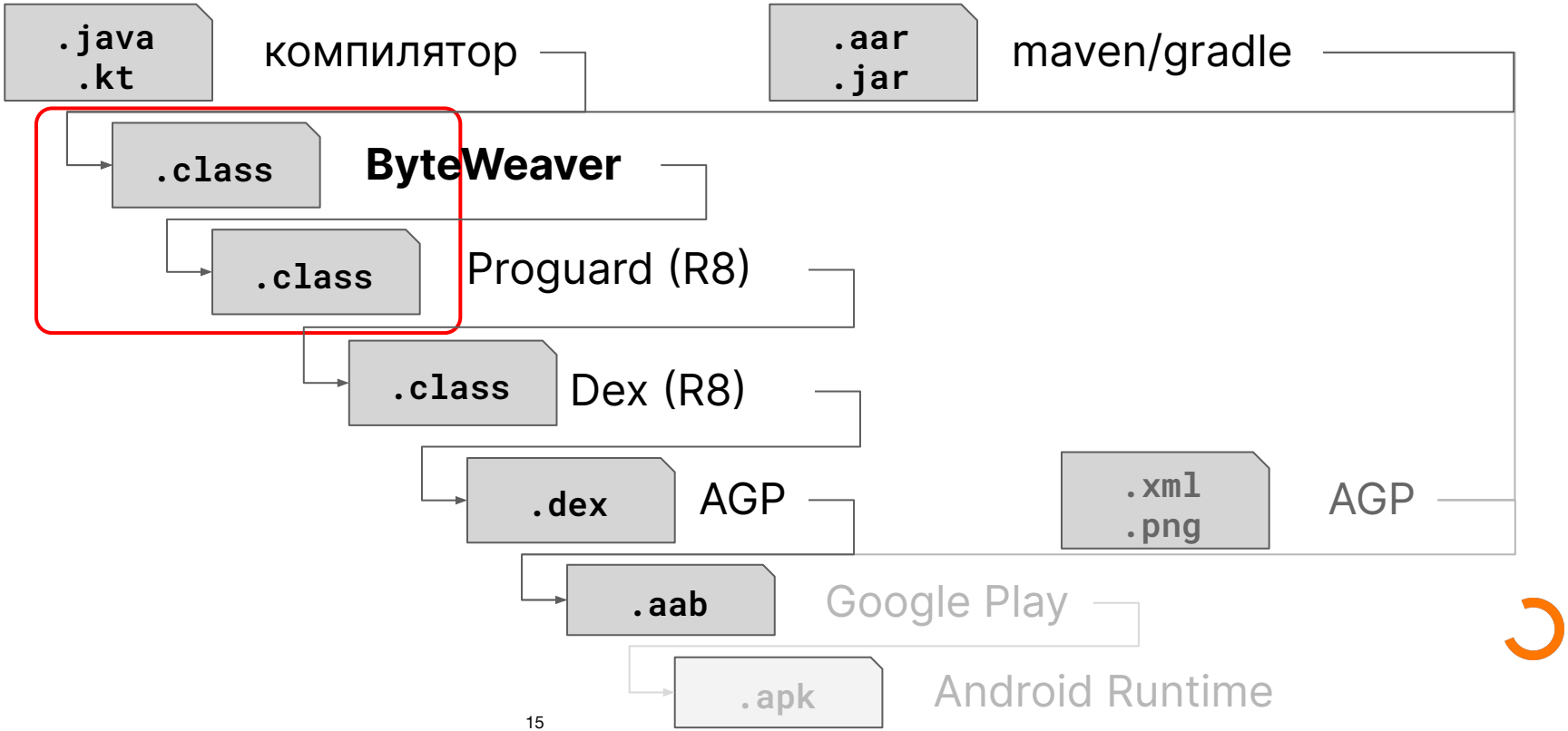
```
.method public final execute(Ljava/lang/Runnable;)I
    .registers 5
    .param p1, "runnable"    # Ljava/lang/Runnable;

    const-string v0, "runnable"

    invoke-static {p1, v0}, Lkotlin/jvm/internal/Intrinsics;→checkNotNullParameter(Ljava/lang/Object;Ljava/lang/String;)V

    .line 7
    nop

    .line 8
    :try_start_6
    const-string v0, "Going to run"

    sget-object v1, Ljava/lang/System;→out:Ljava/io/PrintStream;

    invoke-virtual {v1, v0}, Ljava/io/PrintStream;→println(Ljava/lang/Object;)V

    .line 9
    invoke-interface {p1}, Ljava/lang/Runnable;→run()V
    :try_end_10
    .catchall {:try_start_6 .. :try_end_10} :catchall_11

    goto :goto_19

    .line 10
    :catchall_11
    move-exception v0

    .line 11
    .local v0, "ex":Ljava/lang/Throwable;
    sget-object v1, Ljava/lang/System;→out:Ljava/io/PrintStream;

    const-string v2, "What a Terrible Failure"

    invoke-virtual {v1, v2}, Ljava/io/PrintStream;→println(Ljava/lang/Object;)V

    .line 13
    .end local v0    # "ex":Ljava/lang/Throwable;
    :goto_19
    const/4 v0, 0×0

    return v0
.end method
```

Попався!

OK.TECH

Какой байткод мы можем править, а какой не можем

Когда мы правим байткод



DexClassLoader

Ваш APK **прошел через ByteWeaver**

Ваши классы

модуль приложения

библиотечные модули

Классы из зависимостей

прямых

транзитивных

Системные классы

OK.TECH

Каким образом мы будем его править

Подключим плагин

```
plugins {
    id 'ru.ok.byteweaver' version '0.1.0-rc.2'
}
```

```
pluginManagement {
    repositories {
        maven { url 'https://artifactory-external.vkpartner.ru/artifactory/maven/' }
    }
}
```

Конфигурируем

```
byteweaver {
    debug {
        srcFiles += 'byteweaver/notification-log.conf'
        srcFiles += 'byteweaver/proxy-toast-for-tests.conf'
        srcFiles += 'byteweaver/rx-npe.conf'
    }
    profile {
        srcFiles += 'byteweaver/auto-trace.conf'
    }
    release {
        srcFiles += 'byteweaver/notification-log.conf'
        srcFiles += 'byteweaver/auto-trace.conf'
        srcFiles += 'byteweaver/rx-npe.conf'
    }
}
```

Файлы конфигурации на языке конфигурации ByteWeaver

Определяем классы

```
class io.reactivex.rxjava3.internal.operators.single.SingleFromCallable {

class * extends android.view.View {

class * extends java.lang.Runnable {

class * {

@SomeAnnotation
class ru.ok.android.* {
```

Можем использовать `import`

```
import ru.ok.android.app.NotificationsLogger;
import java.lang.String;
```

Определяем методы

```
class * extends android.app.Activity {
    void onCreate(android.os.Bundle) {

class * extends java.lang.Runnable {
    void run() {

class * {
    @ru.ok.android.commons.os.AutoTraceCompat
    * *(***) {

class * {
    * *(***) {
```

Вставляем код в начало/конец

```
class * {
    @ru.ok.android.commons.os.AutoTraceCompat
    * *(**) {
        before void TraceCompat.beginTraceSection(trace);
        after void TraceCompat.endSection();
    }
}
```

```
class Main {
    @AutoTraceCompat
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

```
class Main {
    @AutoTraceCompat
    public static void main(String[] args) {
        try {
            TraceCompat.beginTraceSection("Main.main(String[])");
            System.out.println("Hello World");
        } finally {
            TraceCompat.endSection();
        }
    }
}
```

Заменяем вызовы

```
class io.reactivex.rxjava3.internal.operators.single.SingleFromCallable {
    * subscribeActual(***) {
        java.lang.Object java.util.concurrent.Callable.call() {
            replace java.lang.Object ru.ok.android.utils.RxNpeChecker.checkCallableCall(self);
        }
    }
}
```

```
public final class SingleFromCallable<T> extends Single<T> {

    final Callable<? extends T> callable;

    public SingleFromCallable(Callable<? extends T> callable) {
        this.callable = callable;
    }

    @Override
    protected void sub
        try {
            Objects.re                                  returned a null value");
        } catch (Throw
            Exceptions
            RxJavaPlugins.onError(ex);
            return;
        }
    }
}
```

callable.call()

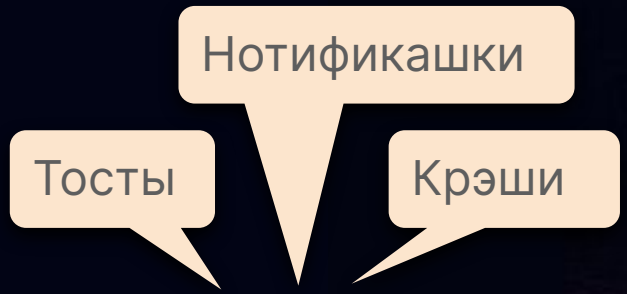
```
public final class SingleFromCallable<T> extends Single<T> {

    final Callable<? extends T> callable;

    public SingleFromCallable(Callable<? extends T> callable) {
        this.callable = callable;
    }

    @Override
    pro
                                                    l value");
        RxJavaPlugins.onError(ex);
        return;
    }
}
```

RxNpeChecker.checkCallableCall(callable)

OK.TECH

Истории из жизни,

их продолжение и окончание



Как поймать тосты

```
class * {
    * *(**) {
        void Toast.show() {
            replace void ToastWatcher.show(self);
        }
    }
}
```

```
object ToastWatcher {
    var listener: (toast: Toast) -> Unit = {}

    @JvmStatic
    fun show(toast: Toast) {
        listener(toast)
        toast.show()
    }
}
```

Это важно!

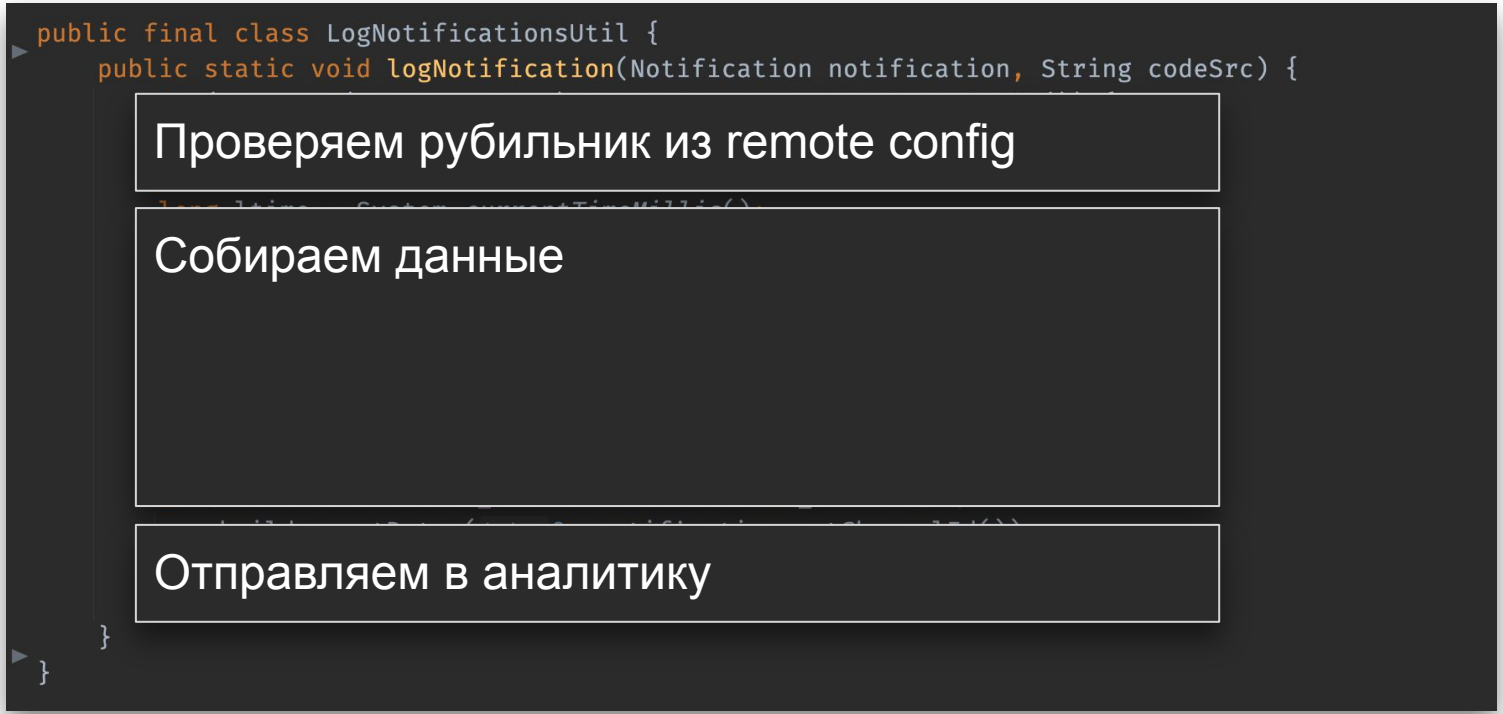
Как залогировать нотификации

```
class * {
    * *(**) {
        void NotificationManager.notify(String, int, Notification) {
            replace void NotificationsLogger.logNotify(self, 0, 1, 2);
        }
    }
}
```

Это важно,
но по-другим
причинам

```
public class NotificationsLogger {

    @KeepName
    public static void logNotify(NotificationManager manager, String tag, int id, Notification notification) {
        LogNotificationsUtil.logNotification(notification, trace());
        manager.notify(tag, id, notification);
    }
}
```

Как залогировать нотификации



Как сначала ничего не понимать...

java.lang.NullPointerException

The callable returned a null value

```
java.util.Objects.requireNonNull(Objects.java:228)
io.reactivex.rxjava3.internal.operators.single.SingleFromCallable.subscribeActual(SingleFromCallable.java:43)
io.reactivex.rxjava3.core.Single.subscribe(Single.java:4855)
io.reactivex.rxjava3.internal.operators.single.SingleSubscribeOn$SubscribeOnObserver.run(SingleSubscribeOn.java:89)
io.reactivex.rxjava3.core.Scheduler$DisposeTask.run(Scheduler.java:644)
io.reactivex.rxjava3.internal.schedulers.ScheduledRunnable.run(ScheduledRunnable.java:65)
io.reactivex.rxjava3.internal.schedulers.ScheduledRunnable.call(ScheduledRunnable.java:56)
java.util.concurrent.FutureTask.run(FutureTask.java:266)
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:301)
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1167)
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:641)
ru.ok.android.utils.NamedThreadFactory$1.run(NamedThreadFactory.java:35)
java.lang.Thread.run(Thread.java:929)
```

Как сначала ничего не понимать...

```java
public final class SingleFromCallable<T> extends Single<T> {

    final Callable<? extends T> callable;

    public SingleFromCallable(Callable<? extends T> callable) {
        this.callable = callable;
    }

    @Override
    protected void subscribeActual(SingleObserver<? super T> observer) {
        try {
            Objects.requireNonNull(callable.call(), message: "The callable returned a null value");
        } catch (Throwable ex) {
            Exceptions.throwIfFatal(ex);
            RxJavaPlugins.onError(ex);
            return;
        }
    }
}
```

Как сначала ничего не понимать...

```
class io.reactivex.rxjava3.internal.operators.single.SingleFromCallable {
    * subscribeActual(***) {
        java.lang.Object java.util.concurrent.Callable.call() {
            replace java.lang.Object ru.ok.android.utils.RxNpeChecker.checkCallableCall(self);
        }
    }
}
```

Дженериков не существует

```
public class RxNpeChecker {
    @SuppressWarnings("unused") // used from generated bytecode
    public static Object checkCallableCall(Callable callable) throws Exception {
        final Object result = callable.call();
        if (result == null) {
            throw new NullPointerException("The callable returned a null value: " + callable);
        }
        return result;
    }
}
```

… а потом ка-а-ак понять!..

java.lang.NullPointerException

The callable returned a null value: l90.b@2f39178

```
ru.ok.android.utils.RxNpeChecker.checkCallableCall(RxNpeChecker.java:12)
io.reactivex.rxjava3.internal.operators.single.SingleFromCallable.subscribeActual(SingleFromCallable.java:43)
io.reactivex.rxjava3.core.Single.subscribe(Single.java:4855)
io.reactivex.rxjava3.internal.operators.single.SingleSubscribeOn$SubscribeOnObserver.run(SingleSubscribeOn.java:89)
io.reactivex.rxjava3.core.Scheduler$DisposeTask.run(Scheduler.java:644)
io.reactivex.rxjava3.internal.schedulers.ScheduledRunnable.run(ScheduledRunnable.java:65)
io.reactivex.rxjava3.internal.schedulers.ScheduledRunnable.call(ScheduledRunnable.java:56)
java.util.concurrent.FutureTask.run(FutureTask.java:264)
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:637)
ru.ok.android.utils.NamedThreadFactory$1.run(NamedThreadFactory.java:35)
java.lang.Thread.run(Thread.java:1012)
```

… а потом ка-а-ак понять!..

```
ru.ok.android.api.rx.core.RxApiClient$$ExternalSyntheticLambda1 -> l90.b:
```

```
@Singleton
public final class RxApiClient {
    @NonNull
    public <T> Single<T> execute(@NonNull ApiExecutableRequest<T> request) {
        return Single.fromCallable(() -> delegate.execute(request))
                .subscribeOn(scheduler);
    }
}
```

… и ка-а-ак починить!..

```java
@Singleton
public final class RxApiClient {
    @NonNull
    public <T> Single<T> execute(@NonNull ApiExecutableRequest<T> request) {
        return Single.fromCallable(() → executeNonNull(request))
                .subscribeOn(scheduler);
    }

    @NonNull
    private <T> T executeNonNull(@NonNull ApiExecutableRequest<T> request) throws IOException, ApiException {
        final T result = delegate.execute(request);
        if (result == null) {
            final String msg = "Parsed api value was null."
                    + " Request: " + request
                    + ", method: " + ApiRequests.extractLogTag(request)
                    + ", parser: " + request.getOkParser();
            throw new NullPointerException(msg);
        }
        return result;
    }
}
```

… и ка-а-ак починить!..

java.lang.NullPointerException

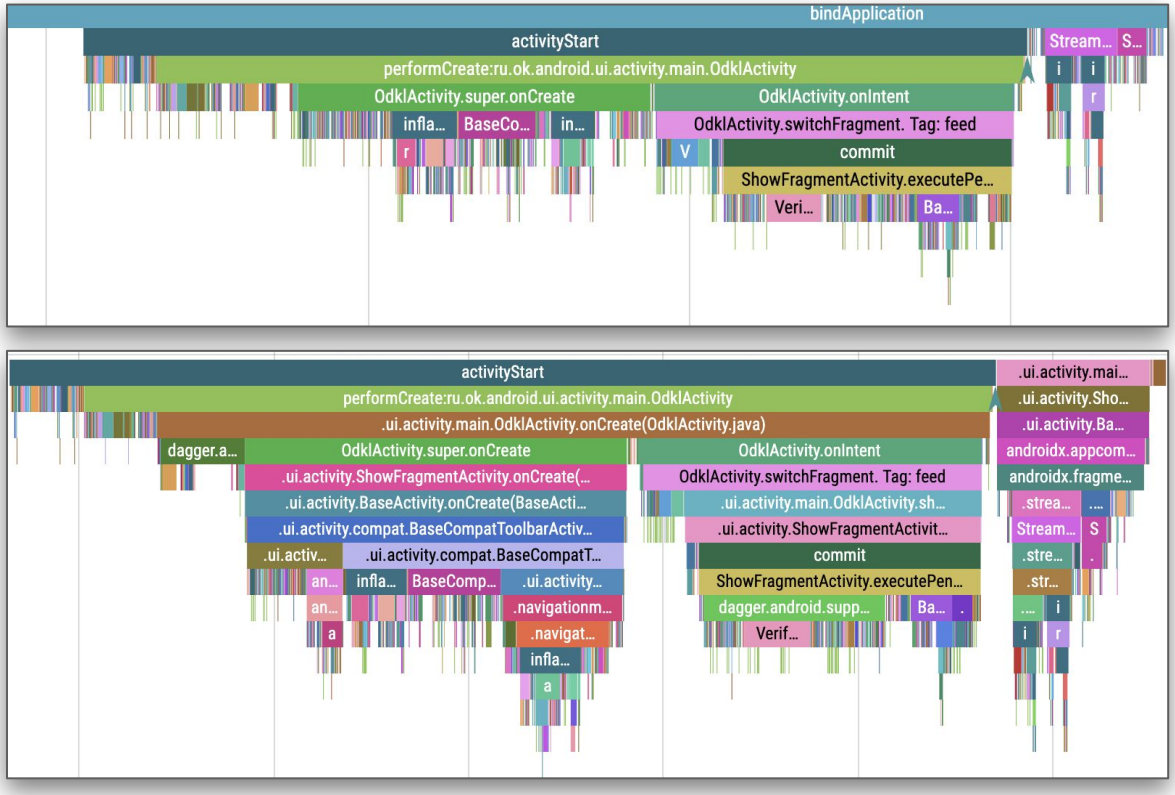
Parsed api value was null. Request: UserInfoRequest{uids=780917803396}, method: users.getInfo, parser: b80.t@43beec0

ru.ok.android.api.rx.core.RxApiClient.executeNonNull(RxApiClient.java:94)
ru.ok.android.api.rx.core.RxApiClient.lambda$execute$0(RxApiClient.java:59)
ru.ok.android.api.rx.core.RxApiClient.$r8$lambda$MM4lebPapqx3ds1rxL9poH2nbac(RxApiClient.java:0)
ru.ok.android.api.rx.core.RxApiClient$$InternalSyntheticLambda1dd62165dfd906473452f39e3ba095757fcd08abc10ab68b78d799e365ed550bc$0.call(RxApiClient.java:4)
ru.ok.android.utils.RxNpeChecker.checkCallableCall(RxNpeChecker.java:8)
io.reactivex.rxjava3.internal.operators.single.SingleFromCallable.subscribeActual(SingleFromCallable.java:43)
io.reactivex.rxjava3.core.Single.subscribe(Single.java:4855)
io.reactivex.rxjava3.internal.operators.single.SingleSubscribeOn$SubscribeOnObserver.run(SingleSubscribeOn.java:89)
io.reactivex.rxjava3.core.Scheduler$DisposeTask.run(Scheduler.java:644)
io.reactivex.rxjava3.internal.schedulers.ScheduledRunnable.run(ScheduledRunnable.java:65)
io.reactivex.rxjava3.internal.schedulers.ScheduledRunnable.call(ScheduledRunnable.java:56)
java.util.concurrent.FutureTask.run(FutureTask.java:264)
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:644)
ru.ok.android.utils.NamedThreadFactory$1.run(NamedThreadFactory.java:35)
java.lang.Thread.run(Thread.java:1012)

Каков улов?

- `RxApiClient`
  - `users.getInfo`
  - `friends.getOnlineV2`
  - `friends.getOutgoingFriendRequests`
  - `friends.invite`
- `LocalPhotoEditorFragment`
- и даже не пришлось форкать RxJava!

Как обогатить SysTrace для Tracer

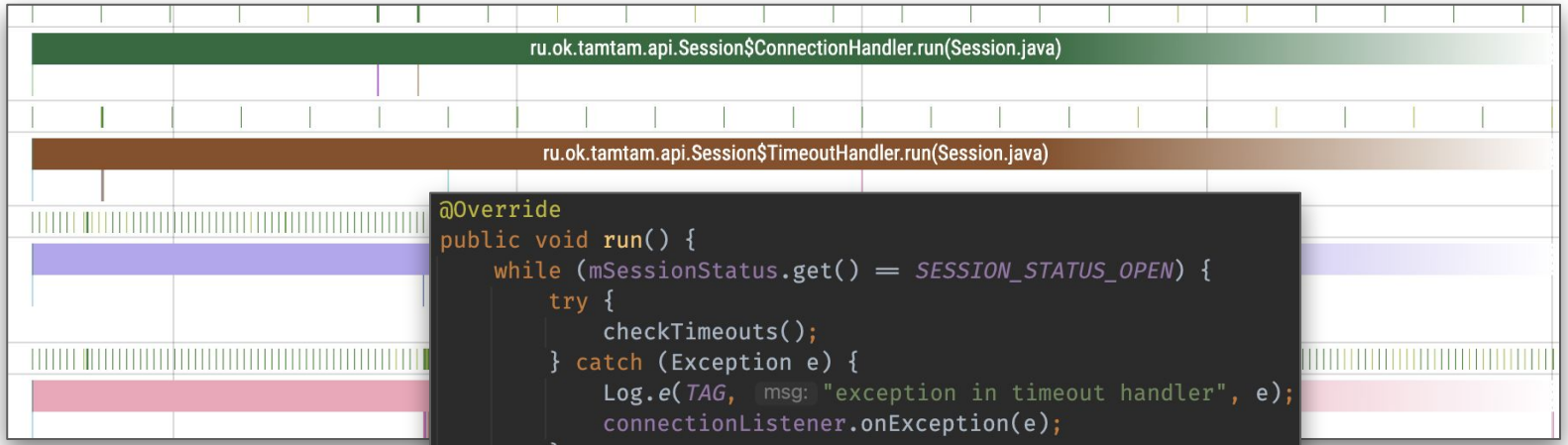


Как обогатить SysTrace для Tracer

```
class *
    @AutoTraceCompat * *(***)

class * extends Activity
    onCreate(Bundle)
    onStart()
    onResume()
    onPause()
    onDestroy()

class * extends Fra
    onCreate()
    onStart()
    onResume()
    onPause()
    onDestroy()
    onCreateView(LayoutInflater, ViewGroup, Bundle)
    onViewCreated(View, Bundle)

class * extends Service
    onStartCommand(Intent, int, int)
```

```
before void TraceCompat.beginTraceSection(trace);
after void TraceCompat.endSection();
```

```
class * extends ContentProvider
    onCreate()

class * extends View
    onAttachedToWindow()
    onDetachedFromWindow()

    handleMessage(Message)

class * extends Handler.Callback
    handleMessage(Message)

class * extends JobIntentService
    onHandleWork(Intent)

class * extends Runnable
    void run()
```




```java
@Override
public void run() {
    while (mSessionStatus.get() == SESSION_STATUS_OPEN) {
        try {
            checkTimeouts();
        } catch (Exception e) {
            Log.e(TAG, msg: "exception in timeout handler", e);
            connectionListener.onException(e);
        }
        try {
            Thread.sleep( millis: 1000);
        } catch (InterruptedException ex) {
            return;
        }
    }
}
```

А что со временем сборки?

+ 5 сек

- Это на большом проекте
- Инкрементальная сборка работает

OK.TECH

Выводы

Чему мы научились

Что мы делали

- Добавляли логи
  - logcat
  - tracer
- Добавляли трейсы
  - systrace
  - tracer
- Приоткрыли черный ящик (для тестов)
- Искали и находили баги

Что могли бы

- Добавить еще какой мониторинг
- Определять (живой) мертвый код
- Ваши варианты

Чего не стали делать, и вам не сто́ит

- Править баги
  - Это неочевидно
  - Это 😡#$!@ в стектрейсе
  - Это 😡#$!@ при дебаге
  - Это увеличивает bus factor
- Генерить “продуктовый” код
  - Трогали только “побочный” код

Что мы поняли

- Иногда нужно копнуть глубже
- Не всегда мы можем подправить исходный код
- ByteWeaver — специфический инструмент для специфических задач
- Копаться в байткоде — это очень увлекательно!

OK.TECH

Планы на будущее

Планы на будущее

- before method args (ro/rw)
- after method result (rw)
- replace body (args, result)
- stopship
- немного декомпиляции

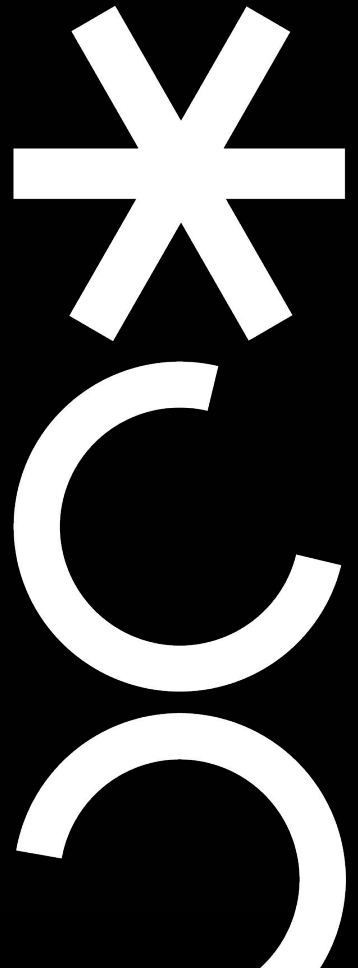
  `log(x)` → `log("x = $x")`

  (главное вовремя остановиться, Саша!!!)
- go opensource

OK.TECH

Спасибо за внимание!

Задавайте вопросы



Александр Асанов, Android разработчик в OK, Tracer, ByteWeaver