

Как сделать своп движка звонков в легаси-коде и не выгореть

Владислав Кожушко





Про меня



Про меня

- С 2012 интересуюсь Android



Про меня

- С 2012 интересуюсь Android
- 3+ года тружусь над звонками



Про меня

- С 2012 интересуюсь Android
- 3+ года тружусь над звонками
- Несу радость в мир 🙌

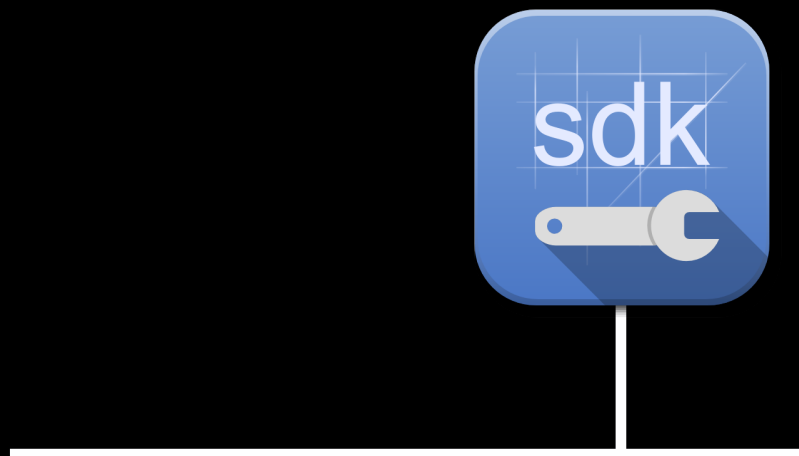
ДВИЖОК

ДВИЖОК

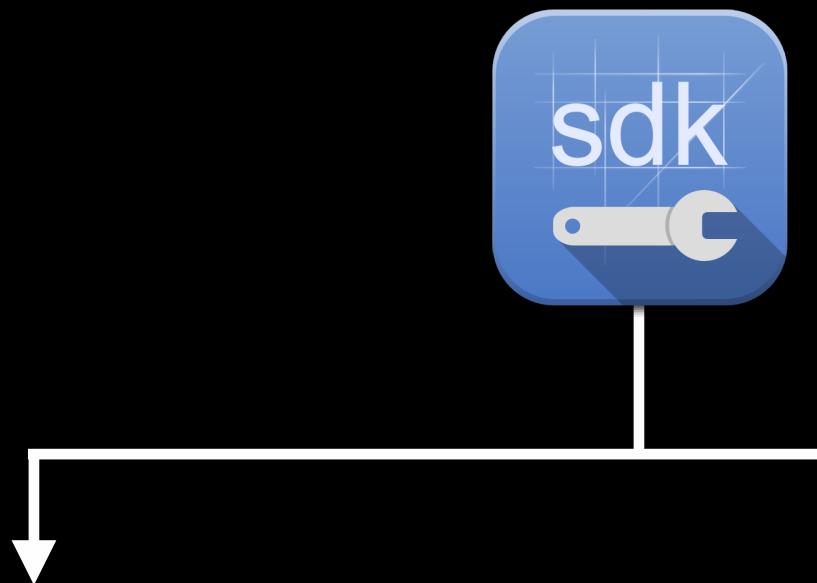
ДВИЖОК



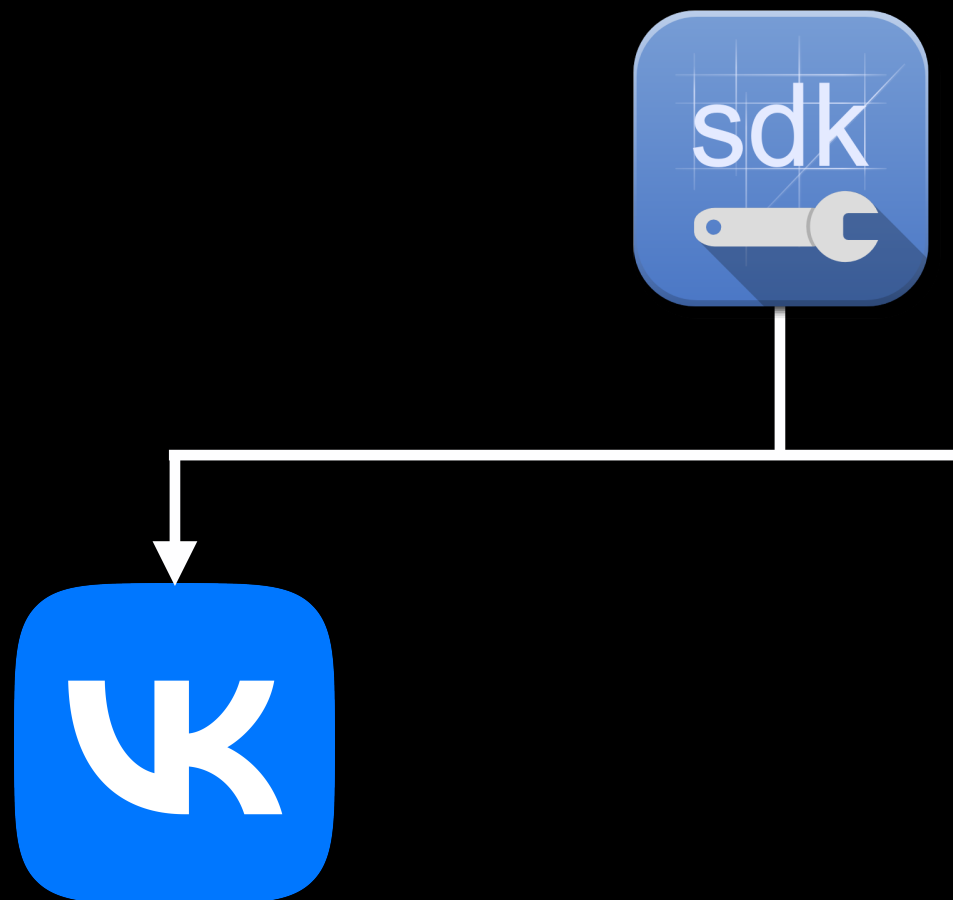
ДВИЖОК



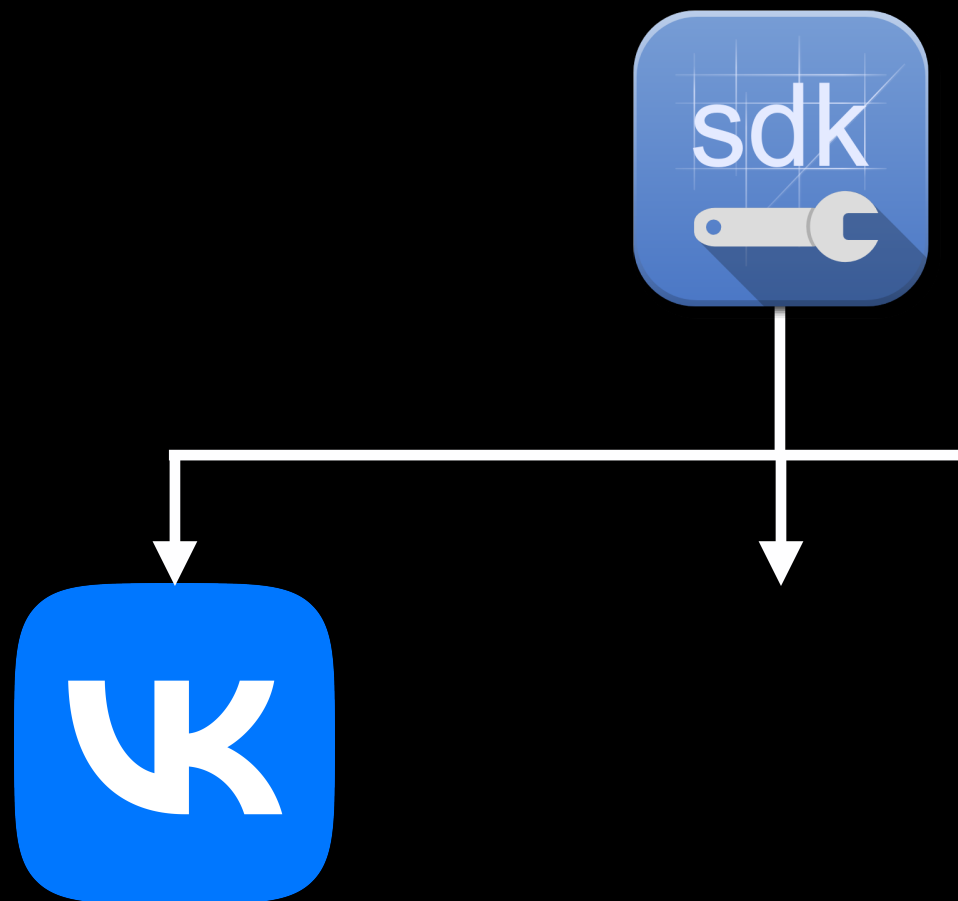
ДВИЖОК



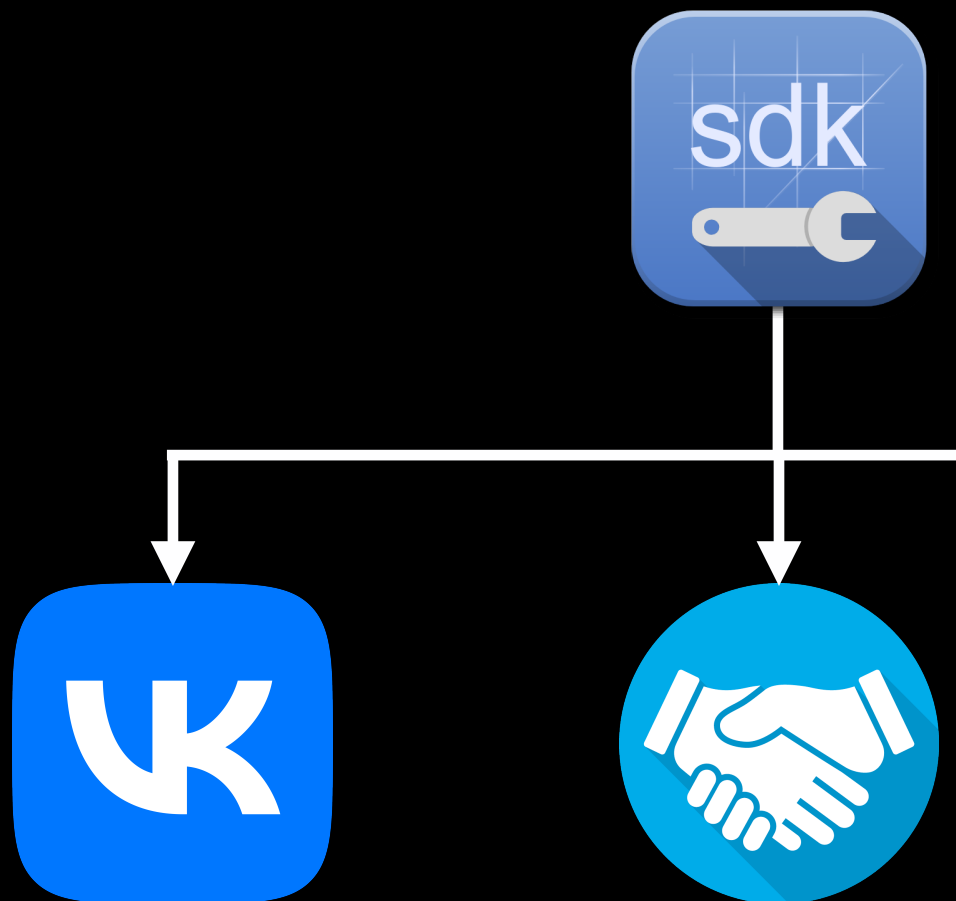
ДВИЖОК



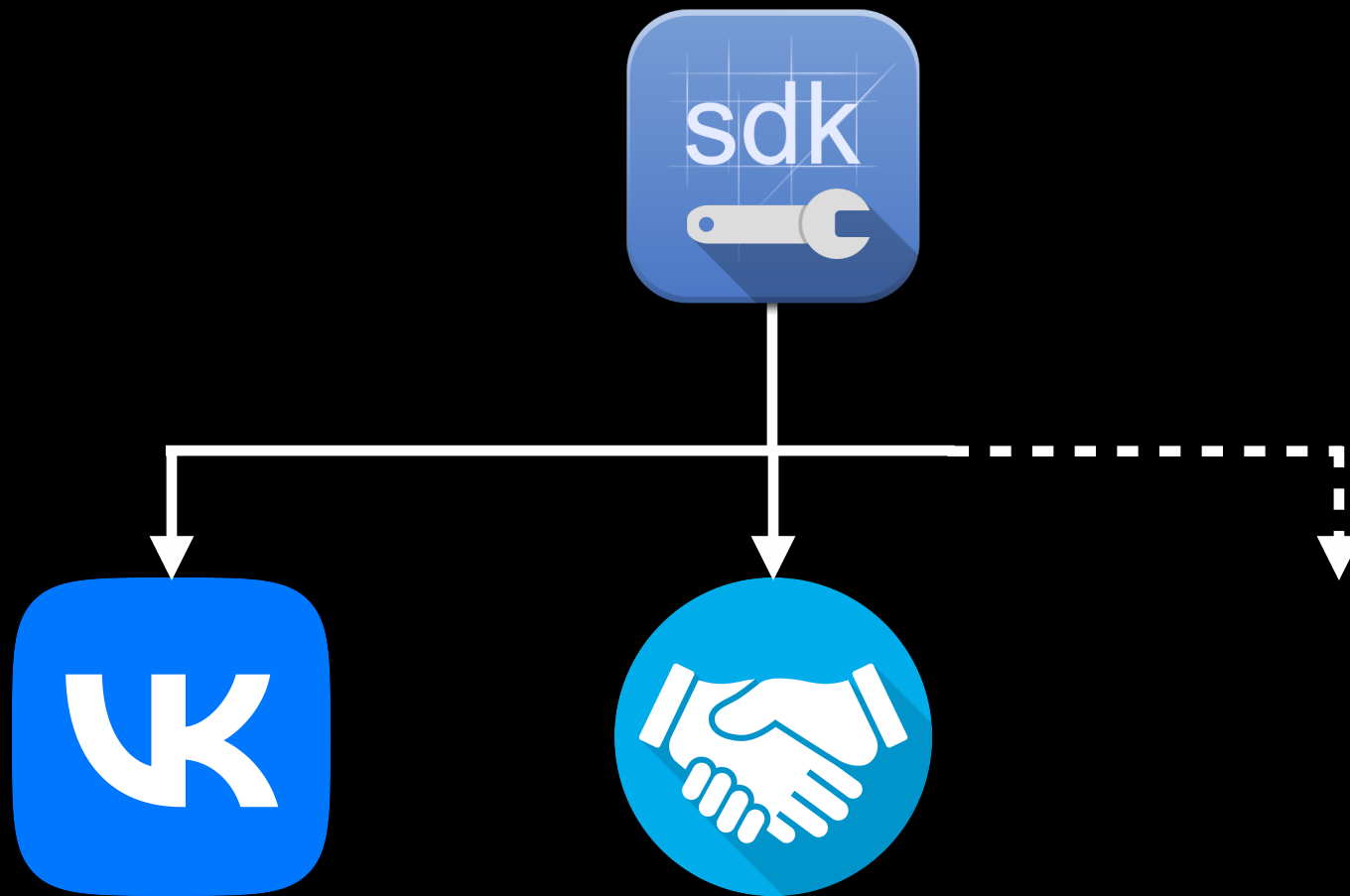
ДВИЖОК



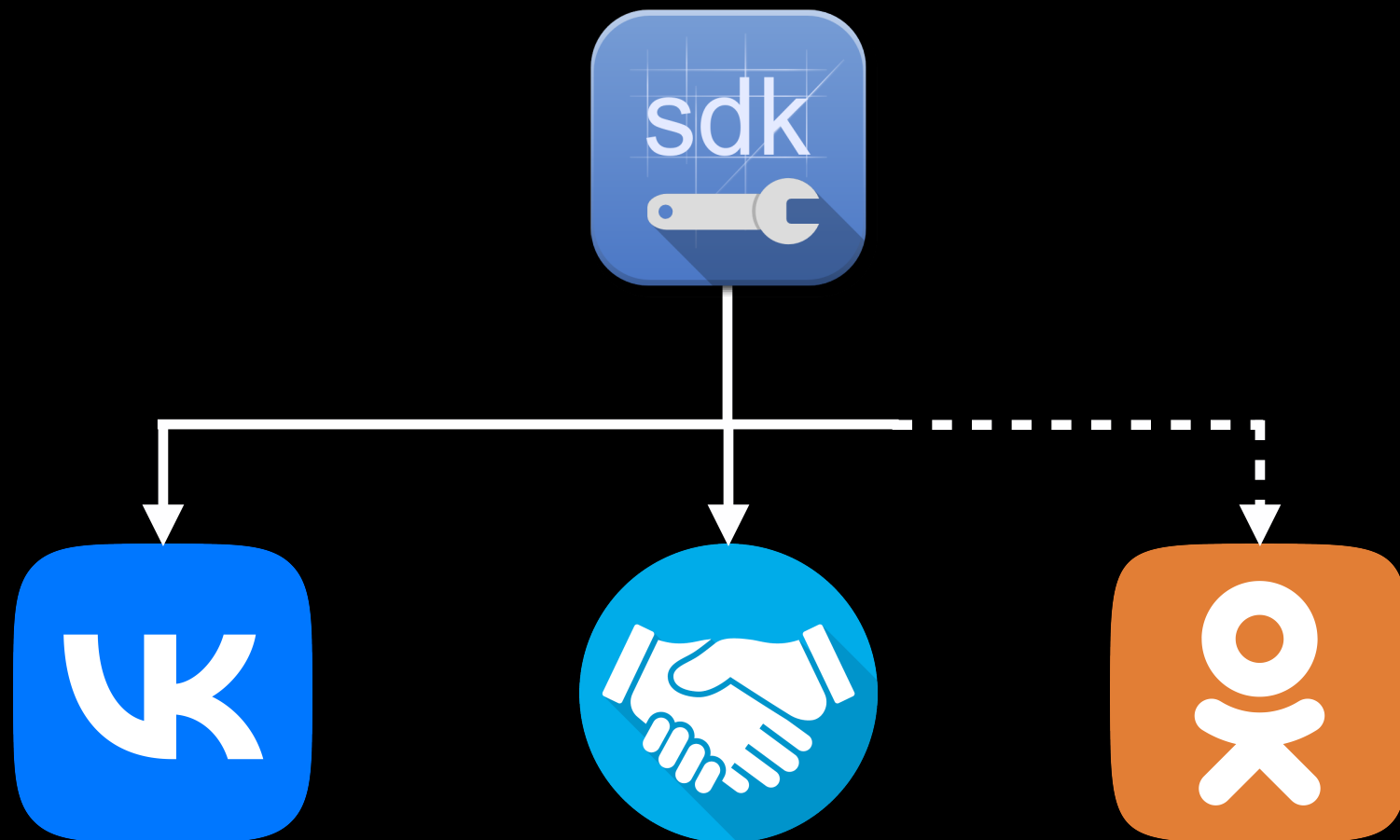
ДВИЖОК



ДВИЖОК



ДВИЖОК



SDK ЗВОHKOB

SDK ЗВОНКОВ

SDK ЗВОНКОВ

- База - WebRTC

SDK ЗВОНКОВ

- База - WebRTC
- ОК инфраструктура

SDK ЗВОНКОВ

- База - WebRTC
- ОК инфраструктура
- Режимы: не ОК, ОК (новый)

SDK ЗВОНКОВ

- База - WebRTC
- ОК инфраструктура
- Режимы: не ОК, ОК (новый)
- ExternalId, InternalId участника

SDK ЗВОНКОВ

- База - WebRTC
- ОК инфраструктура
- Режимы: не ОК, ОК (новый)
- ExternalId, InternalId участника

Работа с id участника:

- ОК API,
- Signaling,
- Topology (p2p, server)

SDK ЗВОНКОВ

SDK ЗВОНКОВ

- ExternalId, InternalId

SDK ЗВОНКОВ

- ExternalId, InternalId

```
1 data class ExternalId(  
2     val id: String,  
3     val isAnon: Boolean  
4 )
```

SDK ЗВОНКОВ

- ExternalId, InternalId

```
1 data class ExternalId(  
2     val id: String,  
3     val isAnon: Boolean  
4 )  
5  
6 data class InternalId(  
7     val id: Long,  
8     val type: Type  
9 ) {  
10  
11     enum class Type(  
12         val value: Char  
13     ) {  
14         USER('u'),  
15         GROUP('g')  
16     }  
17  
18 }
```

SDK ЗВОНКОВ

- ExternalId, InternalId
- Модули: calls-sdk, webrtc-android-sdk, wssignaling

```
1 data class ExternalId(  
2     val id: String,  
3     val isAnon: Boolean  
4 )  
5  
6 data class InternalId(  
7     val id: Long,  
8     val type: Type  
9 ) {  
10  
11     enum class Type(  
12         val value: Char  
13     ) {  
14         USER('u'),  
15         GROUP('g')  
16     }  
17  
18 }
```

SDK ЗВОНКОВ

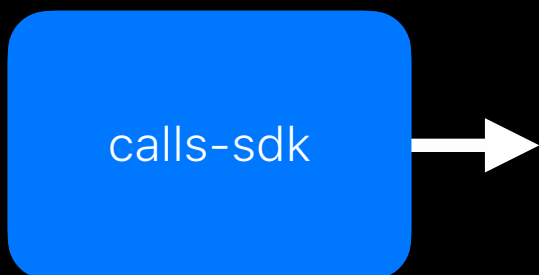
- ExternalId, InternalId
- Модули: calls-sdk, webrtc-android-sdk, wssignaling

calls-sdk

```
1 data class ExternalId(  
2     val id: String,  
3     val isAnon: Boolean  
4 )  
5  
6 data class InternalId(  
7     val id: Long,  
8     val type: Type  
9 ) {  
10  
11     enum class Type(  
12         val value: Char  
13     ) {  
14         USER('u'),  
15         GROUP('g')  
16     }  
17  
18 }
```

SDK ЗВОНКОВ

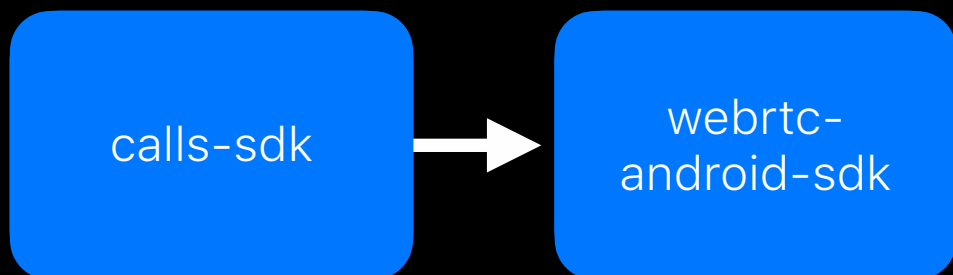
- ExternalId, InternalId
- Модули: calls-sdk, webrtc-android-sdk, wssignaling



```
1 data class ExternalId(  
2     val id: String,  
3     val isAnon: Boolean  
4 )  
5  
6 data class InternalId(  
7     val id: Long,  
8     val type: Type  
9 ) {  
10  
11     enum class Type(  
12         val value: Char  
13     ) {  
14         USER('u'),  
15         GROUP('g')  
16     }  
17  
18 }
```

SDK ЗВОНКОВ

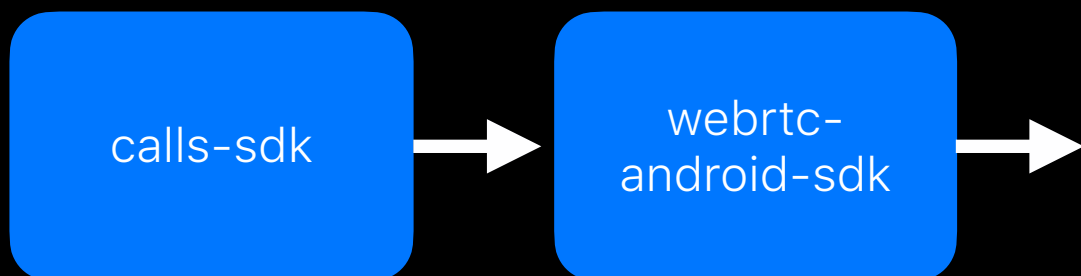
- ExternalId, InternalId
- Модули: calls-sdk, webrtc-android-sdk, wssignaling



```
1 data class ExternalId(  
2     val id: String,  
3     val isAnon: Boolean  
4 )  
5  
6 data class InternalId(  
7     val id: Long,  
8     val type: Type  
9 ) {  
10  
11     enum class Type(  
12         val value: Char  
13     ) {  
14         USER('u'),  
15         GROUP('g')  
16     }  
17  
18 }
```

SDK ЗВОНКОВ

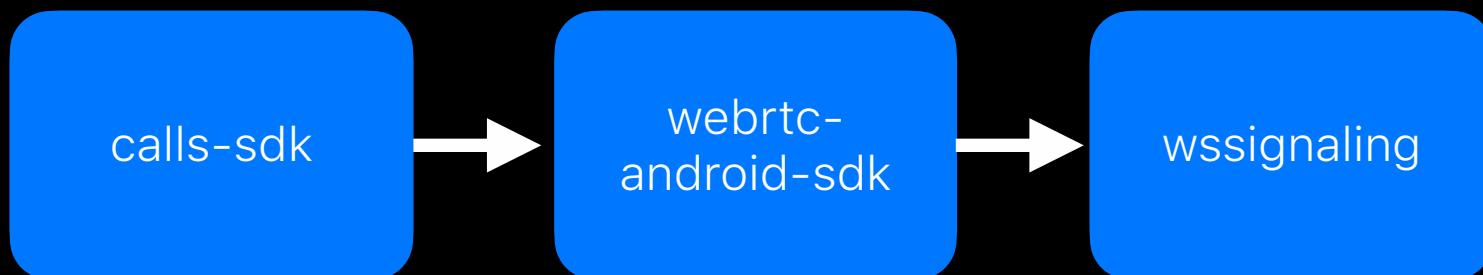
- ExternalId, InternalId
- Модули: calls-sdk, webrtc-android-sdk, wssignaling



```
1 data class ExternalId(  
2     val id: String,  
3     val isAnon: Boolean  
4 )  
5  
6 data class InternalId(  
7     val id: Long,  
8     val type: Type  
9 ) {  
10  
11     enum class Type(  
12         val value: Char  
13     ) {  
14         USER('u'),  
15         GROUP('g')  
16     }  
17  
18 }
```


SDK ЗВОНКОВ

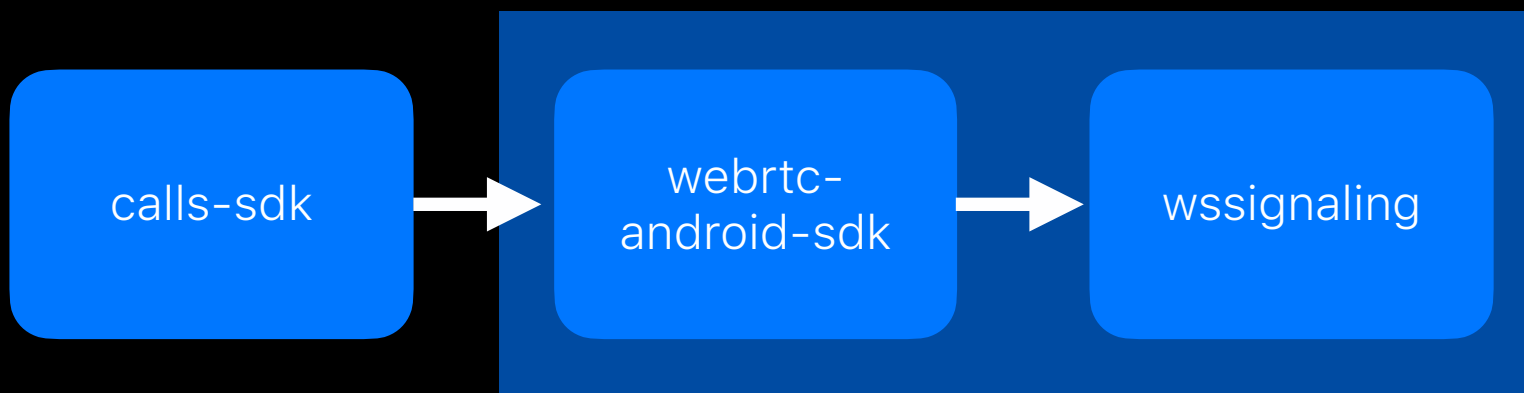
- ExternalId, InternalId
- Модули: calls-sdk, webrtc-android-sdk, wssignaling



```
1 data class ExternalId(  
2     val id: String,  
3     val isAnon: Boolean  
4 )  
5  
6 data class InternalId(  
7     val id: Long,  
8     val type: Type  
9 ) {  
10  
11     enum class Type(  
12         val value: Char  
13     ) {  
14         USER('u'),  
15         GROUP('g')  
16     }  
17  
18 }
```

SDK ЗВОНКОВ

- ExternalId, InternalId
- Модули: calls-sdk, webrtc-android-sdk, wssignaling



```
1 data class ExternalId(  
2     val id: String,  
3     val isAnon: Boolean  
4 )  
5  
6 data class InternalId(  
7     val id: Long,  
8     val type: Type  
9 ) {  
10  
11     enum class Type(  
12         val value: Char  
13     ) {  
14         USER('u'),  
15         GROUP('g')  
16     }  
17  
18 }
```

Ожидание

Ожидание

Ожидание



Поднимаем
версии
библиотек

Ожидание

1

Поднимаем
версии
библиотек

2

Используем
новое API и
модели

Ожидание

1

Поднимаем
версии
библиотек

2

Используем
новое API и
модели

3

???

Ожидание

1

Поднимаем
версии
библиотек

2

Используем
новое API и
модели

3

???

4

Profit

Реальность

Реальность



План

План

План

- Знакомство с проектом и подготовка

План

- Знакомство с проектом и подготовка
- Удаление старого движка

План

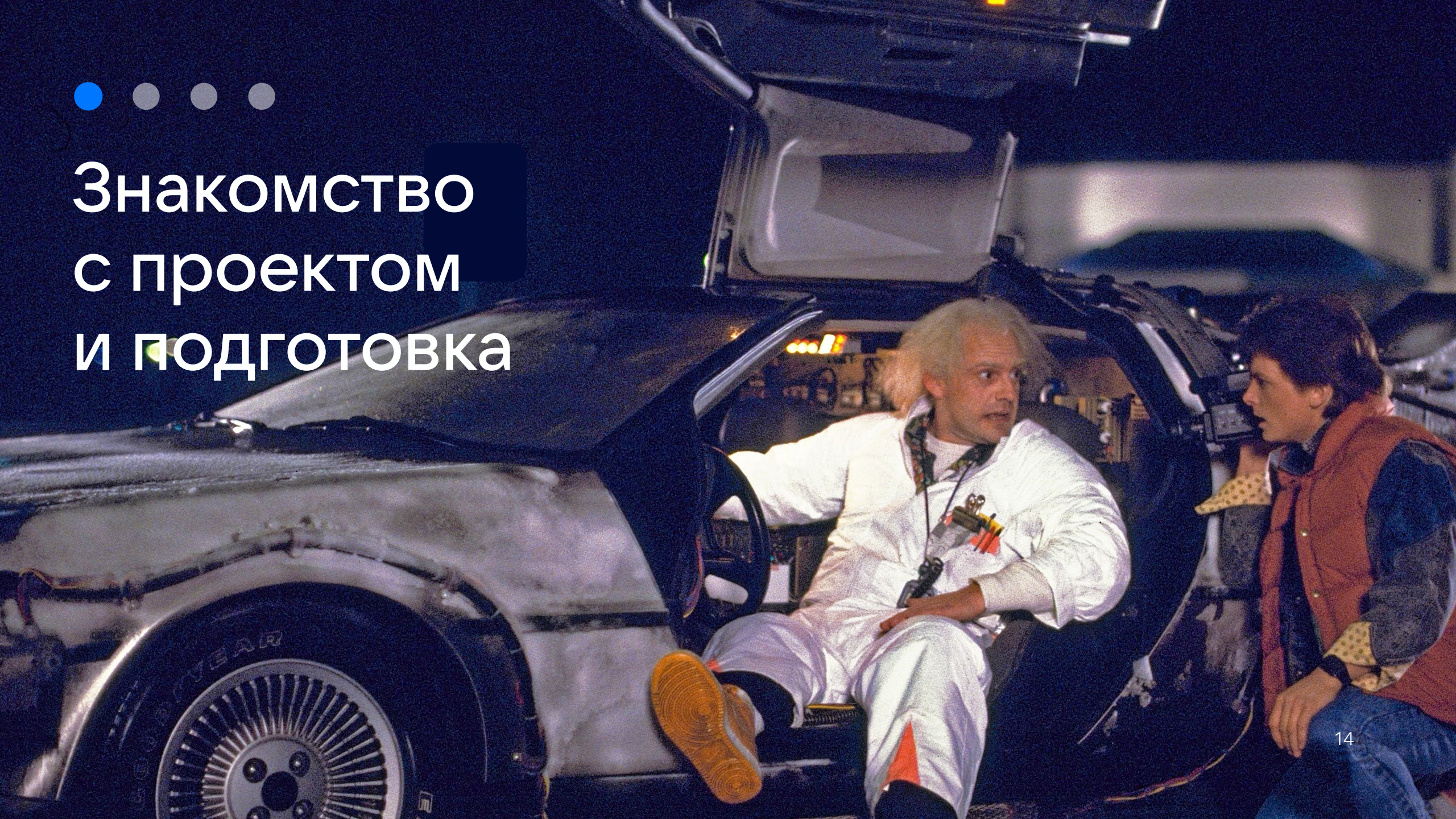
- Знакомство с проектом и подготовка
- Удаление старого движка
- Реализация делегатов

План

- Знакомство с проектом и подготовка
- Удаление старого движка
- Реализация делегатов
- Стабилизация



Знакомство с проектом и подготовка



Знакомство с проектом и подготовка

Знакомство с проектом и подготовка

- Исправление багов, падений

Знакомство с проектом и подготовка

- Исправление багов, падений
- Исправление предупреждений Google

Знакомство с проектом и подготовка

- Исправление багов, падений
- Исправление предупреждений Google
- Уведомления

Знакомство с проектом и подготовка

- Исправление багов, падений
- Исправление предупреждений Google
- Уведомления
- Foreground Service - Android 8+

Знакомство с проектом и подготовка

- Исправление багов, падений
- Исправление предупреждений Google
- Уведомления
- Foreground Service - Android 8+

android.app.RemoteServiceException

Context.startForegroundService() did not then call Service.startForeground(): ServiceRecord{993b5be u0 ru.ok.android/.calls.CallService}

Знакомство с проектом и подготовка

- Исправление багов, падений
- Исправление предупреждений Google
- Уведомления
- Foreground Service - Android 8+






android.app.RemoteServiceException

Context.startForegroundService() did not then call Service.startForeground(): ServiceRecord{993b5be u0 ru.ok.android/.calls.CallService}




Примеры кода

Примеры кода




Примеры кода

```
508 btnAnswerVideo = style.addButton(bottomButtons, color: 0xFF0FD331, iPressedColor: 0xFF09801D, btnSize);
509
510  Drawable i = getResources().getDrawable(ru.ok.android.icons.R.drawable.ic_videocam_32).mutate();
511 i.setColorFilter(Color.WHITE, PorterDuff.Mode.SRC_IN);
512 btnAnswerVideo.setImageDrawable(i);
513
514  Drawable i2 = getResources().getDrawable(ru.ok.android.icons.R.drawable.ico_phone_32).mutate();
515 i2.setColorFilter(Color.BLACK, PorterDuff.Mode.SRC_IN);
516 btnAnswerAudio.setImageDrawable(i2);
517
518  Drawable i3 = getResources().getDrawable(ru.ok.android.icons.R.drawable.ico_phone_off_32).mutate();
519 i3.setColorFilter(Color.WHITE, PorterDuff.Mode.SRC_IN);
520 btnDecline.setImageDrawable(i3);
```




Примеры кода

```
508 btnAnswerVideo = style.addButton(bottomButtons, color: 0xFF0FD331, iPressedColor: 0xFF09801D, btnSize);
509
510  Drawable i = getResources().getDrawable(ru.ok.android.icons.R.drawable.ic_videocam_32).mutate();
511 i.setColorFilter(Color.WHITE, PorterDuff.Mode.SRC_IN);
512 btnAnswerVideo.setImageDrawable(i);
513
514  Drawable i2 = getResources().getDrawable(ru.ok.android.icons.R.drawable.ico_phone_32).mutate();
515 i2.setColorFilter(Color.BLACK, PorterDuff.Mode.SRC_IN);
516 btnAnswerAudio.setImageDrawable(i2);
517
518  Drawable i3 = getResources().getDrawable(ru.ok.android.icons.R.drawable.ico_phone_off_32).mutate();
519 i3.setColorFilter(Color.WHITE, PorterDuff.Mode.SRC_IN);
520 btnDecline.setImageDrawable(i3);
```




Примеры кода

```
508 btnAnswerVideo = style.addButton(bottomButtons, color: 0xFF0FD331, iPressedColor: 0xFF09801D, btnSize);
509
510  Drawable i = getResources().getDrawable(ru.ok.android.icons.R.drawable.ic_videocam_32).mutate();
511 i.setColorFilter(Color.WHITE, PorterDuff.Mode.SRC_IN);
512 btnAnswerVideo.setImageDrawable(i);
513
514  Drawable i2 = getResources().getDrawable(ru.ok.android.icons.R.drawable.ico_phone_32).mutate();
515 i2.setColorFilter(Color.BLACK, PorterDuff.Mode.SRC_IN);
516 btnAnswerAudio.setImageDrawable(i2);
517
518  Drawable i3 = getResources().getDrawable(ru.ok.android.icons.R.drawable.ico_phone_off_32).mutate();
519 i3.setColorFilter(Color.WHITE, PorterDuff.Mode.SRC_IN);
520 btnDecline.setImageDrawable(i3);
```

Примеры кода

```
508 btnAnswerVideo = style.addButton(bottomButtons, color: 0xFF0FD331, iPressedColor: 0xFF09801D, btnSize);
509
510  Drawable i = getResources().getDrawable(ru.ok.android.icons.R.drawable.ic_videocam_32).mutate();
511 i.setColorFilter(Color.WHITE, PorterDuff.Mode.SRC_IN);
512 btnAnswerVideo.setImageDrawable(i);
513
514  Drawable i2 = getResources().getDrawable(ru.ok.android.icons.R.drawable.ico_phone_32).mutate();
515 i2.setColorFilter(Color.BLACK, PorterDuff.Mode.SRC_IN);
516 btnAnswerAudio.setImageDrawable(i2);
517
518  Drawable i3 = getResources().getDrawable(ru.ok.android.icons.R.drawable.ico_phone_off_32).mutate();
519 i3.setColorFilter(Color.WHITE, PorterDuff.Mode.SRC_IN);
520 btnDecline.setImageDrawable(i3);
```

Примеры кода

```
508 btnAnswerVideo = style.addButton(bottomButtons, color: 0xFF0FD331, iPressedColor: 0xFF09801D, btnSize);
509
510  Drawable i = getResources().getDrawable(ru.ok.android.icons.R.drawable.ic_videocam_32).mutate();
511 i.setColorFilter(Color.WHITE, PorterDuff.Mode.SRC_IN);
512 btnAnswerVideo.setImageDrawable(i);
513
514  Drawable i2 = getResources().getDrawable(ru.ok.android.icons.R.drawable.ico_phone_32).mutate();
515 i2.setColorFilter(Color.BLACK, PorterDuff.Mode.SRC_IN);
516 btnAnswerAudio.setImageDrawable(i2);
517
518  Drawable i3 = getResources().getDrawable(ru.ok.android.icons.R.drawable.ico_phone_off_32).mutate();
519 i3.setColorFilter(Color.WHITE, PorterDuff.Mode.SRC_IN);
520 btnDecline.setImageDrawable(i3);
```

Примеры кода

Примеры кода

```
266     public OKCall(final @NonNull Context appCtx,  
267                 @NonNull final CallParticipant currentUserCallParticipant,  
268                 final @NonNull CallTarget target,  
269                 final boolean caller,  
270                 final @NonNull String cid,  
271                 final @Nullable RTCLog log,  
272                 final @Nullable RTCCallParams rtcCallParams,  
273                 final @NonNull String place,  
274                 final boolean hasVideo,  
275                 boolean newEmptyCall,  
276                 boolean isAnonym,  
277                 boolean createJoinLink,  
278                 @Nullable String joinLink,  
279                 @NonNull UserInfo currentUser,  
280                 final RTCStatistics rtcStat,  
281                 @NonNull UxCallManager callManager,  
282                 @NonNull CallsNotificator callsNotificator  
283     ) {
```

Примеры кода

```
266     public OKCall(final @NonNull Context appCtx,  
267                 @NonNull final CallParticipant currentUserCallParticipant,  
268                 final @NonNull CallTarget target,  
269                 final boolean caller,  
270                 final @NonNull String cid,  
271                 final @Nullable RTCLog log,  
272                 final @Nullable RTCCallParams rtcCallParams,  
273                 final @NonNull String place,  
274                 final boolean hasVideo,  
275                 boolean newEmptyCall,  
276                 boolean isAnonym,  
277                 boolean createJoinLink,  
278                 @Nullable String joinLink,  
279                 @NonNull UserInfo currentUser,  
280                 final RTCStatistics rtcStat,  
281                 @NonNull UxCallManager callManager,  
282                 @NonNull CallsNotificator callsNotificator  
283     ) {
```

17!

Примеры кода

```
266 public OKCall(final @NonNull Context appCtx,  
267               @NonNull final CallParticipant currentUserCallParticipant,  
268               final @NonNull CallTarget target,  
269               final boolean caller,  
270               final @NonNull String cid,  
271               final @Nullable RTCTLog log,  
272               final @Nullable RTCCallParams rtcCallParams,  
273               final @NonNull String place,  
274               final boolean hasVideo,  
275               boolean newEmptyCall,  
276               boolean isAnonym,  
277               boolean createJoinLink,  
278               @Nullable String joinLink,  
279               @NonNull UserInfo currentUser,  
280               final RTCStatistics rtcStat,  
281               @NonNull UxCallManager callManager,  
282               @NonNull CallsNotificator callsNotificator  
283               ) {
```

17!

```
1 fun createP2pCall(params: StartOkCallParams){  
2     conversationFactory.call { builder ->  
3         builder.setOpponentId(params.opponent.toExternalId())  
4                 .setMyId(myId)  
5                 .setStartWithVideo(params.isVideo)  
6                 .setEventListener(params.listeners)  
7                 .setOnPrepared(params.onSuccess)  
8                 .setOnError(params.onError)  
9                 .build()  
10 }
```

Примеры кода

```
64     public CellView(@NonNull final Context context,  
65                     final EglBase.Context eglContext,  
66                     @NonNull final OKCall okCall,  
67                     final CallParticipant.ParticipantId participantId,  
68                     @NonNull final DimenUtils calc  
69     ) {
```

Примеры кода

```
1 /**  
2  * @deprecated Don't even think about it!  
3  */  
4 @Deprecated  
5 public class CallGlobals {  
6     public static ICallPms PMS;  
7     public static ICallRouter ROUTER;  
8     public static ICallArgsFactory CALL_ARGS_FACTORY;  
9     public static IUsersCache USERS_CACHE;  
10 }
```

Примеры кода

Примеры кода

👤 Vlad Kozhushko *

```
sealed class CallTarget {  
    abstract val name: String?  
    abstract val pic: String?
```

👤 Vlad Kozhushko

```
abstract fun targetParam(): Pair<String, String>
```

👤 Vlad Kozhushko

```
open fun hasChat() = false
```

new *

```
class ByConversationId(  
    @JvmField val callId: String,  
) : CallTarget() {  
    override val name = null  
    override val pic = null
```

new *

```
override fun targetParam() = "callId" to callId
```



```
}
```

Примеры кода

👤 Vlad Kozhushko *

```
sealed class CallTarget {  
    abstract val name: String?  
    abstract val pic: String?
```

👤 Vlad Kozhushko

```
abstract fun targetParam(): Pair<String, String>
```

👤 Vlad Kozhushko

```
open fun hasChat() = false
```

new *

```
class ByConversationId(  
    @JvmField val callId: String,  
) : CallTarget() {  
    override val name = null  
    override val pic = null
```

new *

```
override fun targetParam() = "callId" to callId
```



```
}
```


Примеры кода

Примеры кода



```
1 public enum CallEvents {  
2     ICE_CONNECTED,  
3     ICE_DISCONNECTED,  
4  
5     PARTICIPANT_HANGUP,  
6  
7     ACCEPTED_ON_OTHER_DEVICE  
8     //...  
9 }
```

Примеры кода



```
1 public enum CallEvents {  
2     ICE_CONNECTED,  
3     ICE_DISCONNECTED,  
4  
5     PARTICIPANT_HANGUP,  
6  
7     ACCEPTED_ON_OTHER_DEVICE  
8     //...  
9 }
```

calls-sdk



webrtc-android-sdk

Примеры кода

```
1 public enum CallEvents {  
2     ICE_CONNECTED,  
3     ICE_DISCONNECTED,  
4  
5     PARTICIPANT_HANGUP,  
6  
7     ACCEPTED_ON_OTHER_DEVICE  
8     //...  
9 }
```

```
1 @Override  
2 public void onEvent(CallEvents e, Call c, SignalingErrors.GenericError error) {  
3     Logger.d("e: " + e);  
4     CallView view = this;  
5     switch (e) {  
6         case CAMERA_CHANGED:  
7             view.post(this::updatePipRendererMirroring);  
8             break;  
9         case CONVERSATION_CLOSED:  
10            //...  
11    }  
12 }
```

calls-sdk



webrtc-android-sdk





Назад в
2011 год

Док, куда
мы попали?

С чем имеем дело?

С чем имеем
дело?

- Java

С чем имеем
дело?

- Java
- Божественные классы

С чем имеем
дело?

С чем имеем дело?

- Java
- Божественные классы
- Singleton'ы и статические методы

С чем имеем дело?

- Java
- Божественные классы
- Singleton'ы и статические методы
- Архитектура???

С чем имеем дело?

- Java
- Божественные классы
- Singleton'ы и статические методы
- Архитектура???
- AsyncTask

С чем имеем дело?

- Java
- Божественные классы
- Singleton'ы и статические методы
- Архитектура???
- AsyncTask
- "Обфусцированный" код

С чем имеем дело?

- Java
- Божественные классы
- Singleton'ы и статические методы
- Архитектура???
- AsyncTask
- "Обфусцированный" код
- Спагетти код

С чем имеем дело?

- Java
- Божественные классы
- Singleton'ы и статические методы
- Архитектура???
- AsyncTask
- "Обфусцированный" код
- Спагетти код
- Внутренние модули pre-SDK*

С чем имеем дело?

- Java
- Божественные классы
- Singleton'ы и статические методы
- Архитектура???
- AsyncTask
- "Обфусцированный" код
- Спагетти код
- Внутренние модули pre-SDK*
- XOR id пользователей

Итого

Итого

Итого

- Новый проект

Итого

- Новый проект
- Исправлены баги и блокеры

Итого

- Новый проект
- Исправлены баги и блокеры
- Изоляция кода звонков (API-, Impl-модули)

```
1 interface VoipCallsBridge {  
2     fun openCallScreen()  
3     fun joinByLink(paramFun: (JoinCallByLink.Builder) -> JoinCallByLink)  
4     fun joinToCall(paramFun: (JoinCallById.Builder) -> JoinCallById)  
5     //...  
6 }
```


Итого

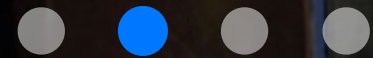
- Новый проект
- Исправлены баги и блокеры
- Изоляция кода звонков (API-, Impl-модули)
- Dagger2

```
1 interface VoipCallsBridge {
2     fun openCallScreen()
3     fun joinByLink(paramFun: (JoinCallByLink.Builder) -> JoinCallByLink)
4     fun joinToCall(paramFun: (JoinCallById.Builder) -> JoinCallById)
5     //...
6 }
```

Итого

- Новый проект
- Исправлены баги и блокеры
- Изоляция кода звонков (API-, Impl-модули)
- Dagger2
- Диета для **View**

```
1 interface VoipCallsBridge {
2     fun openCallScreen()
3     fun joinByLink(paramFun: (JoinCallByLink.Builder) -> JoinCallByLink)
4     fun joinToCall(paramFun: (JoinCallById.Builder) -> JoinCallById)
5     //...
6 }
```



Удаление старой реализации

Сюрприз



Сюрприз

OK RxJava2 -> RxJava3

ВКонтакте RxJava3 + RxJava2 (звонков)

SDK RxJava2

Миграция RxJava2 → RxJava3

Миграция RxJava2 → RxJava3



```
1 //3.0.0
2 public ScheduledDirectTask(Runnable runnable) {
3     super(runnable);
4 }
```

Миграция RxJava2 → RxJava3

```
1 //3.0.0
2 public ScheduledDirectTask(Runnable runnable) {
3     super(runnable);
4 }
```

```
1 //3.1.6
2 public ScheduledDirectTask(Runnable runnable, boolean interruptOnCancel) {
3     super(runnable, interruptOnCancel);
4 }
```


Миграция RxJava2 → RxJava3

- Переименование операторов

```
1 //3.0.0
2 public ScheduledDirectTask(Runnable runnable) {
3     super(runnable);
4 }
```

```
1 //3.1.6
2 public ScheduledDirectTask(Runnable runnable, boolean interruptOnCancel) {
3     super(runnable, interruptOnCancel);
4 }
```

Миграция RxJava2 → RxJava3

- Переименование операторов
- Disposables -> Disposable

```
1 //3.0.0
2 public ScheduledDirectTask(Runnable runnable) {
3     super(runnable);
4 }
```

```
1 //3.1.6
2 public ScheduledDirectTask(Runnable runnable, boolean interruptOnCancel) {
3     super(runnable, interruptOnCancel);
4 }
```

Миграция RxJava2 → RxJava3

- Переименование операторов
- Disposables -> Disposable
- ВКонтакте - перестал работать Rx stream - в версиях 3.1.6 и 3.0.0

```
1 //3.0.0
2 public ScheduledDirectTask(Runnable runnable) {
3     super(runnable);
4 }
```

```
1 //3.1.6
2 public ScheduledDirectTask(Runnable runnable, boolean interruptOnCancel) {
3     super(runnable, interruptOnCancel);
4 }
```

Итого



Итого



Итого

- Делегаты



Итого

- Делегаты
- Подключен SDK



Итого

- Делегаты
- Подключен SDK
- mainframer + includeBuild



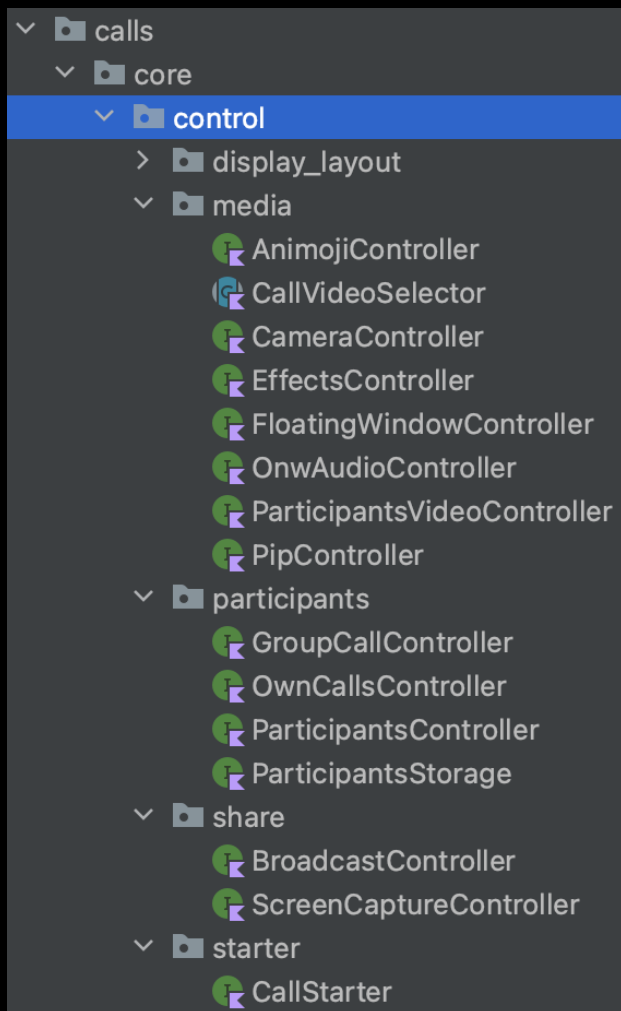
Итого

- Делегаты
- Подключен SDK
- mainframer + includeBuild
- Интерфейсы и реализация

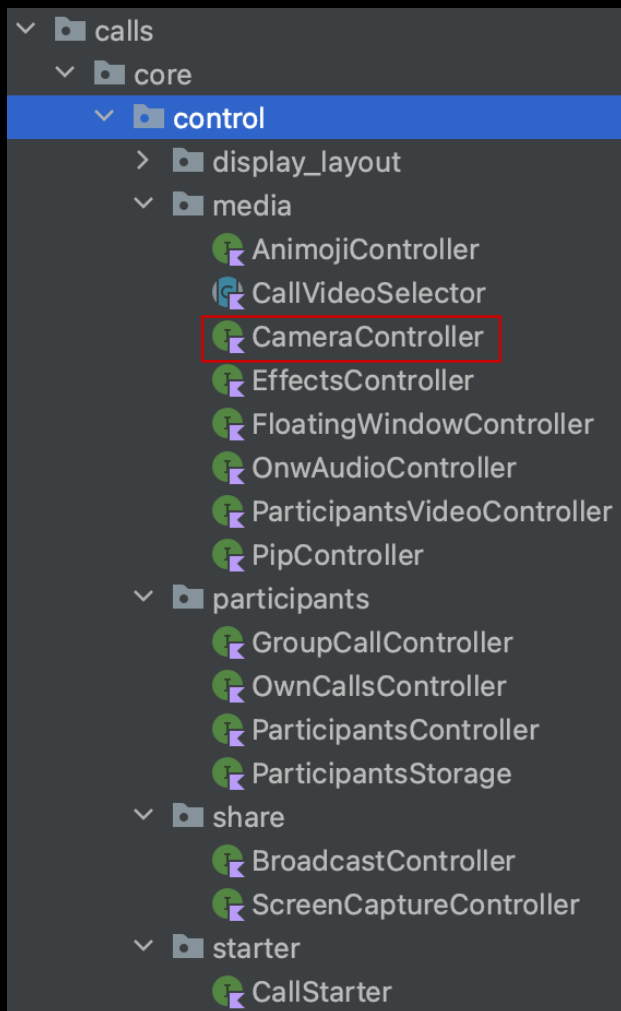
Итого

- Делегаты
- Подключен SDK
- mainframer + includeBuild
- Интерфейсы и реализация
- Заглушки - проект собирается, но ничего не делает

Делегаты



Делегаты



```
1 interface CameraController {
2     val numberOfCameras: Int
3
4     val isFrontCamera: Boolean
5     var isVideoEnabled: Boolean
6
7     fun swapCamera()
8 }
```



Реализация функциональности

Реализация функциональности

Реализация функциональности

1. Правки SDK под ОК

Реализация функциональности

1. Правки SDK под ОК
2. Другой ЖЦ звонка

Реализация функциональности

1. Правки SDK под ОК
2. Другой ЖЦ звонка
3. Реализация фич звонка (делегатов)

Реализация функциональности

1. Правки SDK под ОК
2. Другой ЖЦ звонка
3. Реализация фич звонка (делегатов)
4. Решение проблем

Правки SDK. Сетевой слой

Правки SDK. Сетевой слой

```
1 // 4 конструктора
2 public class OkApi {
3     @NonNull
4     private final SimpleApiClient apiClient;
5     @NonNull
6     private final RxApiClient rxApiClient;
7     @NonNull
8     private final ExternApiConfigProvider apiConfigProvider;
9
10    private final ApiClientEngine sourceApiClient;
11    private ApiRequestDebugger debugger = ApiRequestDebugger.NO_OP;
12
13 }
```

Правки SDK. Сетевой слой


```
1 // 4 конструктора
2 public class OkApi {
3     @NonNull
4     private final SimpleApiClient apiClient;
5     @NonNull
6     private final RxApiClient rxApiClient;
7     @NonNull
8     private final ExternApiConfigProvider apiConfigProvider;
9
10    private final ApiClientEngine sourceApiClient;
11    private ApiRequestDebugger debugger = ApiRequestDebugger.NO_OP;
12
13 }
```

```
1 interface OkApi {
2     val apiClient: ApiClient
3     val rxApiClient: RxApiClient
4     val okApiHolder: OkApiHolder
5
6     val builder: Builder
7
8     open class Builder {
9         //...
10        open fun build(): OkApi = DefaultOkApi(this)
11    }
12 }
```

Правки SDK. Сетевой слой

```
1 // 4 конструктора
2 public class OkApi {
3     @NonNull
4     private final SimpleApiClient apiClient;
5     @NonNull
6     private final RxApiClient rxApiClient;
7     @NonNull
8     private final ExternApiConfigProvider apiConfigProvider;
9
10    private final ApiClientEngine sourceApiClient;
11    private ApiRequestDebugger debugger = ApiRequestDebugger.NO_OP;
12
13 }
```

```
1 interface OkApi {
2     val apiClient: ApiClient
3     val rxApiClient: RxApiClient
4     val okApiHolder: OkApiHolder
5
6     val builder: Builder
7
8     open class Builder {
9         //...
10        open fun build(): OkApi = DefaultOkApi(this)
11    }
12 }
```



Правки SDK. Сетевой слой

```
1 // 4 конструктора
2 public class OkApi {
3     @NonNull
4     private final SimpleApiClient apiClient;
5     @NonNull
6     private final RxApiClient rxApiClient;
7     @NonNull
8     private final ExternApiConfigProvider apiConfigProvider;
9
10    private final ApiClientEngine sourceApiClient;
11    private ApiRequestDebugger debugger = ApiRequestDebugger.NO_OP;
12
13 }
```

```
1 interface OkApi {
2     val apiClient: ApiClient
3     val rxApiClient: RxApiClient
4     val okApiHolder: OkApiHolder
5
6     val builder: Builder
7
8     open class Builder {
9         //...
10        open fun build(): OkApi = DefaultOkApi(this)
11    }
12 }
```

```
1 internal class DefaultOkApi(
2     override val builder: OkApi.Builder
3 ) : OkApi {
4
5     override val apiClient: ApiClient
6     override val rxApiClient: RxApiClient
7     override val okApiHolder: OkApiHolder
8
9     init {
10        //как-то это инициализируется
11    }
12 }
```

Правки SDK. Сетевой слой

```
1 // 4 конструктора
2 public class OkApi {
3     @NonNull
4     private final SimpleApiClient apiClient;
5     @NonNull
6     private final RxApiClient rxApiClient;
7     @NonNull
8     private final ExternApiConfigProvider apiConfigProvider;
9
10    private final ApiClientEngine sourceApiClientEngine;
11    private ApiRequestDebugger debugger = ApiRequestDebugger.NO_OP;
12
13 }
```

```
1 interface OkApi {
2     val apiClient: ApiClient
3     val rxApiClient: RxApiClient
4     val okApiHolder: OkApiHolder
5
6     val builder: Builder
7
8     open class Builder {
9         //...
10        open fun build(): OkApi = DefaultOkApi(this)
11    }
12 }
```

```
1 internal class DefaultOkApi(
2     override val builder: OkApi.Builder
3 ) : OkApi {
4
5     override val apiClient: ApiClient
6     override val rxApiClient: RxApiClient
7     override val okApiHolder: OkApiHolder
8
9     init {
10        //как-то это инициализируется
11    }
12 }
```


Правки SDK. Сетевой слой

```
1 // 4 конструктора
2 public class OkApi {
3     @NonNull
4     private final SimpleApiClient apiClient;
5     @NonNull
6     private final RxApiClient rxApiClient;
7     @NonNull
8     private final ExternApiConfigProvider apiConfigProvider;
9
10    private final ApiClientEngine sourceApiEngine;
11    private ApiRequestDebugger debugger = ApiRequestDebugger.NO_OP;
12
13 }
```

```
1 interface OkApi {
2     val apiClient: ApiClient
3     val rxApiClient: RxApiClient
4     val okApiHolder: OkApiHolder
5
6     val builder: Builder
7
8     open class Builder {
9         //...
10        open fun build(): OkApi = DefaultOkApi(this)
11    }
12 }
```

```
1 internal class PortalOkApi(
2     override val builder: Builder
3 ) : OkApi {
4
5     override val apiClient: ApiClient
6     override val rxApiClient: RxApiClient
7     override val okApiHolder = object : OkApiHolder {}
8
9     class Builder : OkApi.Builder() {
10        //...
11        override fun build() = PortalOkApi(this)
12    }
13
14    init {
15        //и тут тоже
16    }
17 }
```

```
1 internal class DefaultOkApi(
2     override val builder: OkApi.Builder
3 ) : OkApi {
4
5     override val apiClient: ApiClient
6     override val rxApiClient: RxApiClient
7     override val okApiHolder: OkApiHolder
8
9     init {
10        //как-то это инициализируется
11    }
12 }
```

Правки SDK. Сетевой слой

```
1 // 4 конструктора
2 public class OkApi {
3     @NonNull
4     private final SimpleApiClient apiClient;
5     @NonNull
6     private final RxApiClient rxApiClient;
7     @NonNull
8     private final ExternApiConfigProvider apiConfigProvider;
9
10    private final ApiClientEngine sourceApiEngine;
11    private ApiRequestDebugger debugger = ApiRequestDebugger.NO_OP;
12
13 }
```

```
1 interface OkApiHolder {
2     fun markSessionExpired() {}
3     fun clearSession() {}
4     fun setRequestDebugger(debugger: ApiRequestDebugger) {}
5 }
```

```
1 interface OkApi {
2     val apiClient: ApiClient
3     val rxApiClient: RxApiClient
4     val okApiHolder: OkApiHolder
5
6     val builder: Builder
7
8     open class Builder {
9         //...
10        open fun build(): OkApi = DefaultOkApi(this)
11    }
12 }
```

```
1 internal class PortalOkApi(
2     override val builder: Builder
3 ) : OkApi {
4
5     override val apiClient: ApiClient
6     override val rxApiClient: RxApiClient
7     override val okApiHolder = object : OkApiHolder {}
8
9     class Builder : OkApi.Builder() {
10        //...
11        override fun build() = PortalOkApi(this)
12    }
13
14    init {
15        //и тут тоже
16    }
17 }
```

```
1 internal class DefaultOkApi(
2     override val builder: OkApi.Builder
3 ) : OkApi {
4
5     override val apiClient: ApiClient
6     override val rxApiClient: RxApiClient
7     override val okApiHolder: OkApiHolder
8
9     init {
10        //как-то это инициализируется
11    }
12 }
```

Правки SDK. Логирование

Правки SDK. Логирование

```
implementation ("ru.ok.android:calls-sdk:${rootProject['calls.sdk.version']}") {  
    exclude group: 'ru.ok.android', module: 'onelogger'  
}
```

Правки SDK. Логирование

```
implementation ("ru.ok.android:calls-sdk:${rootProject['calls.sdk.version']}") {  
    exclude group: 'ru.ok.android', module: 'onelogger'  
}
```

```
init {  
    try {  
        OneLogger.ensureInitialized()  
    } catch (ignored: NoClassDefFoundError) {  
        // костыль для ОК пока что будем делать exclude OneLogger в ОК  
    }  
}
```

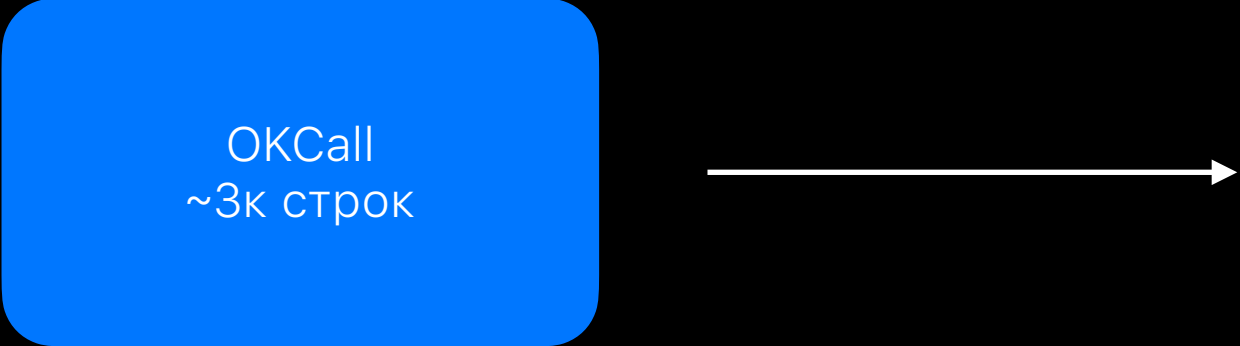
Реализация делегатов

Реализация делегатов

OKCall
~3к строк

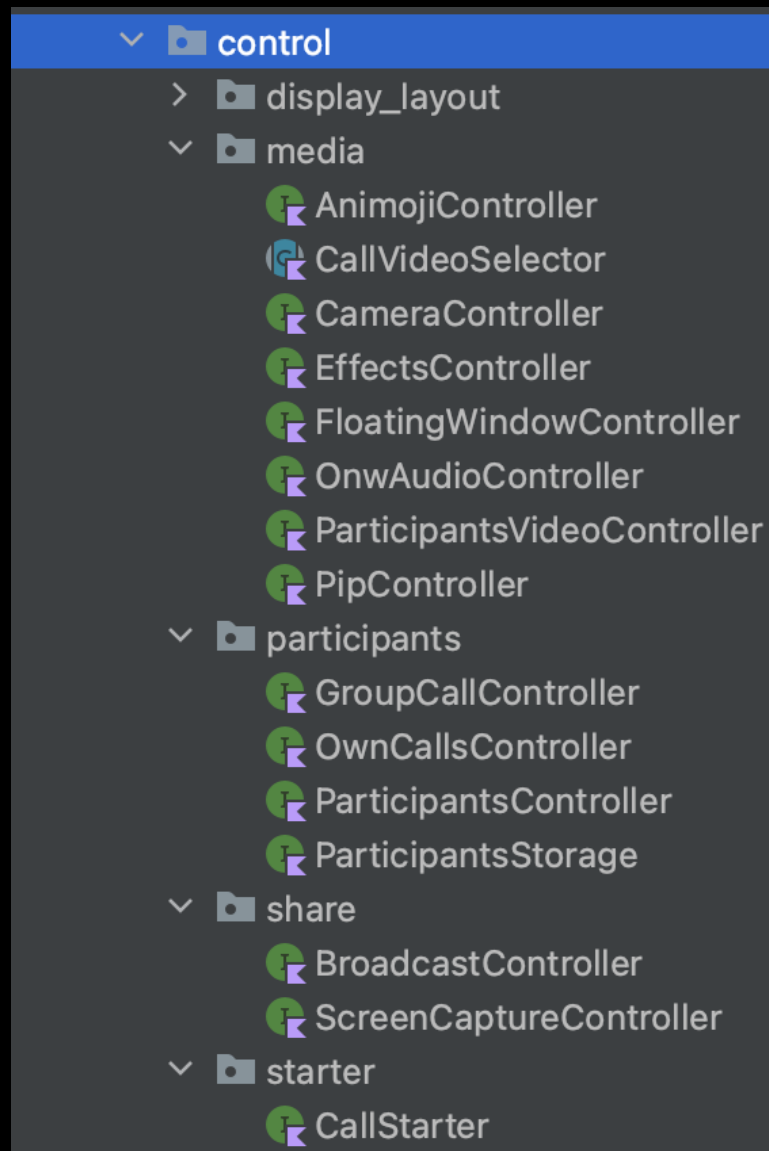
Реализация делегатов

OKCall
~3к строк



Реализация делегатов

OKCall
~3к строк



Старт звонка

Старт звонка

```
1 public OKCall(  
2     final @NonNull Context appCtx,  
3     final @NonNull CallParticipant currentCallUser,  
4     final @NonNull CallTarget target,  
5     final boolean caller,  
6     final @NonNull String cid,  
7     final @Nullable RTCLog log,  
8     final @Nullable RTCCallParams rtcCallParams,  
9     final @NonNull String place,  
10    final boolean hasVideo,  
11    boolean newEmptyCall,  
12    boolean isAnonym,  
13    boolean createJoinLink,  
14    @Nullable String joinLink,  
15    @NonNull UserInfo currentUser,  
16    final RTCStatistics rtcStat,  
17    @NonNull UxCallManager callManager,  
18    @NonNull CallsNotificator callsNotificator) {  
19    //~200 строк, много if  
20 }
```

Старт звонка

```
1 public OKCall(  
2     final @NonNull Context appCtx,  
3     final @NonNull CallParticipant currentCallUser,  
4     final @NonNull CallTarget target,  
5     final boolean caller,  
6     final @NonNull String cid,  
7     final @Nullable RTCLog log,  
8     final @Nullable RTCCallParams rtcCallParams,  
9     final @NonNull String place,  
10    final boolean hasVideo,  
11    boolean newEmptyCall,  
12    boolean isAnonym,  
13    boolean createJoinLink,  
14    @Nullable String joinLink,  
15    @NonNull UserInfo currentUser,  
16    final RTCStatistics rtcStat,  
17    @NonNull UxCallManager callManager,  
18    @NonNull CallsNotificator callsNotificator) {  
19    //~200 строк, много if  
20 }
```



Старт звонка

```
1 public OKCall(  
2     final @NonNull Context appCtx,  
3     final @NonNull CallParticipant currentCallUser,  
4     final @NonNull CallTarget target,  
5     final boolean caller,  
6     final @NonNull String cid,  
7     final @Nullable RTCLog log,  
8     final @Nullable RTCCallParams rtcCallParams,  
9     final @NonNull String place,  
10    final boolean hasVideo,  
11    boolean newEmptyCall,  
12    boolean isAnonym,  
13    boolean createJoinLink,  
14    @Nullable String joinLink,  
15    @NonNull UserInfo currentUser,  
16    final RTCStatistics rtcStat,  
17    @NonNull UxCallManager callManager,  
18    @NonNull CallsNotificator callsNotificator) {  
19    //~200 строк, много if  
20 }
```



```
1 internal class ConversationKreator @Inject constructor(  
2     private val conversationFactory: ConversationFactory,  
3 ) {  
4  
5     private fun callOneToOne(  
6         params: StartOkCallParams,  
7         onSuccess: (Conversation) -> Unit,  
8         onError: (Throwable) -> Unit  
9     ): Conversation = conversationFactory.call { builder ->  
10         builder.setOpponentId(params.opponentId)  
11             .setMyId(params.myId)  
12             .setStartWithVideo(params.isVideo)  
13             .setEventListener(params.listeners)  
14             .setOnPrepared(onSuccess)  
15             .setOnError(onError)  
16             .build()  
17     }  
18  
19 }
```

Работа с участниками

Работа с участниками



```
1 interface ParticipantsStorage {  
2     val me: ParticipantPair  
3     val opponent: ParticipantPair?  
4  
5     val participantsUpdates: Observable<ParticipantsState>  
6  
7     val participants: Collection<ParticipantPair>  
8     //  
9 }
```

Работа с участниками



```
1 interface ParticipantsStorage {  
2     val me: ParticipantPair  
3     val opponent: ParticipantPair?  
4  
5     val participantsUpdates: Observable<ParticipantsState>  
6  
7     val participants: Collection<ParticipantPair>  
8     //  
9 }
```



```
1 data class ParticipantPair(  
2     val member: CallMember,  
3     val user: OkCallUser  
4 ) : CallMember by member, OkCallUser by user
```


Работа с участниками

```
1 interface ParticipantsStorage {  
2     val me: ParticipantPair  
3     val opponent: ParticipantPair?  
4  
5     val participantsUpdates: Observable<ParticipantsState>  
6  
7     val participants: Collection<ParticipantPair>  
8     //  
9 }
```

```
1 data class ParticipantPair(  
2     val member: CallMember,  
3     val user: OkCallUser  
4 ) : CallMember by member, OkCallUser by user
```

```
1 interface CallMember {  
2     val id: CallMemberId  
3  
4     val isAudioEnabled: VideoState  
5     val isVideoEnabled: VideoState  
6     val screenCaptureState: VideoState  
7  
8     val isConnected: Bool  
9     val isAccepted: Bool  
10    val acceptCallEpochMs: Long  
11    val isSelf: Bool  
12 }
```

Работа с участниками

```
1 interface ParticipantsStorage {  
2     val me: ParticipantPair  
3     val opponent: ParticipantPair?  
4  
5     val participantsUpdates: Observable<ParticipantsState>  
6  
7     val participants: Collection<ParticipantPair>  
8     //  
9 }
```

```
1 data class ParticipantPair(  
2     val member: CallMember,  
3     val user: OkCallUser  
4 ) : CallMember by member, OkCallUser by user
```

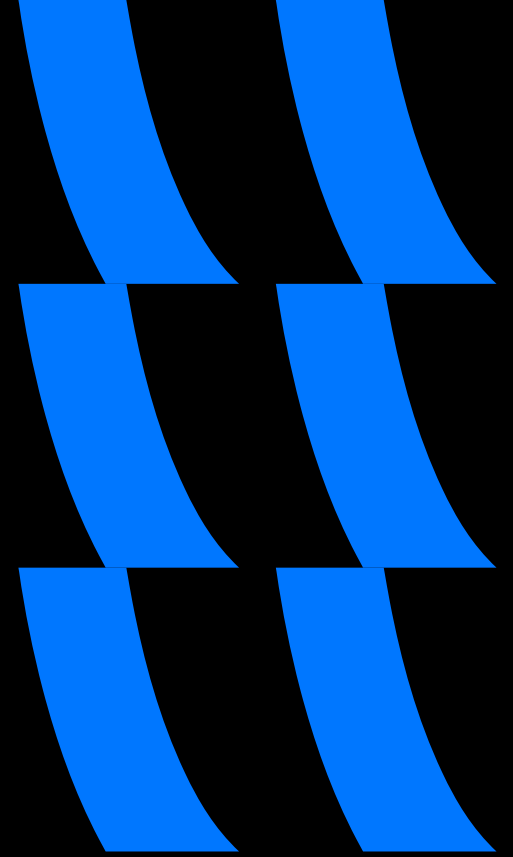
```
1 interface CallMember {  
2     val id: CallMemberId  
3  
4     val isAudioEnabled: VideoState  
5     val isVideoEnabled: VideoState  
6     val screenCaptureState: VideoState  
7  
8     val isConnected: Bool  
9     val isAccepted: Bool  
10    val acceptCallEpochMs: Long  
11    val isSelf: Bool  
12 }
```

```
1 interface OkCallUser {  
2     val uid: CallUserId  
3     val name: String  
4     val picUrl: String?  
5     val isGroup: Bool  
6     val isAnon: Bool  
7 }
```

НОВЫЙ сюрприз

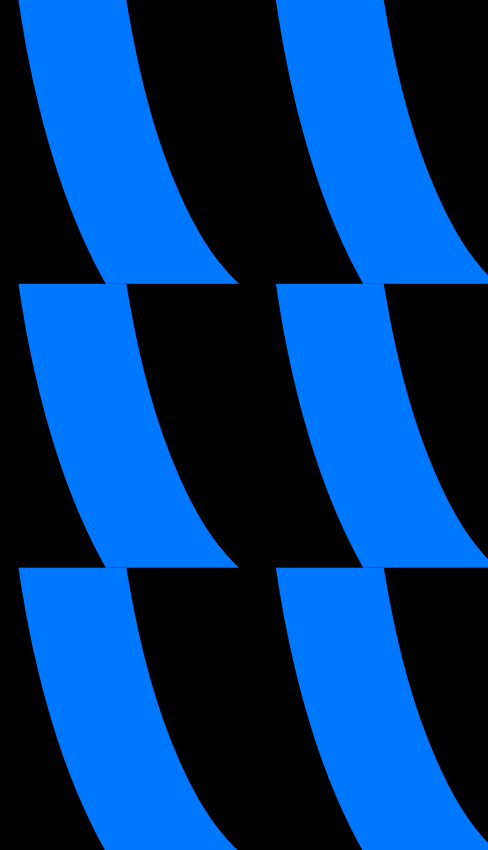


Java 17, AGP 8



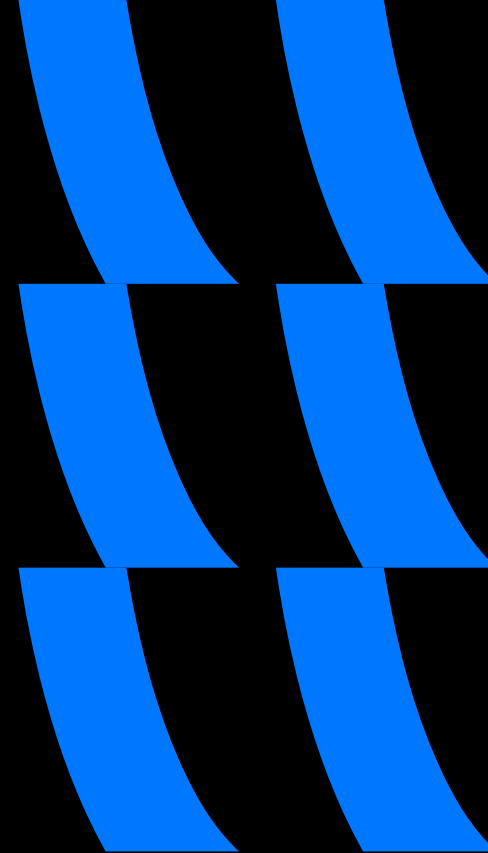
Java 17, AGP 8

- Правка build.gradle(.kts)



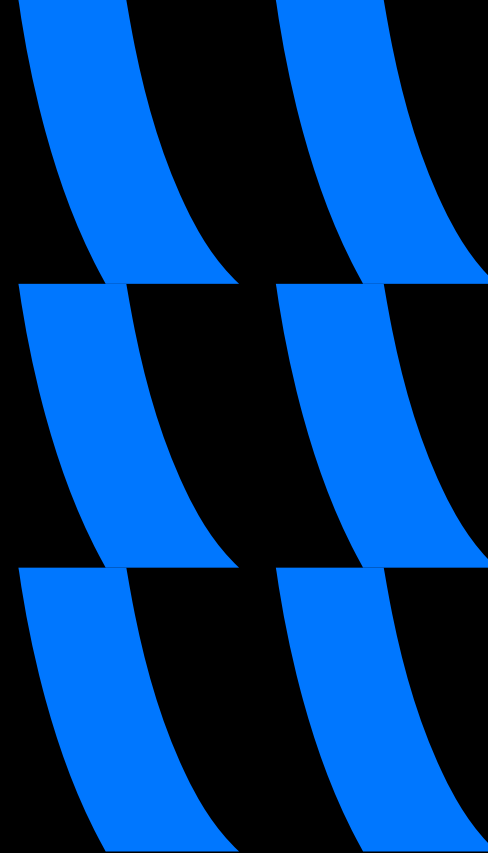
Java 17, AGP 8

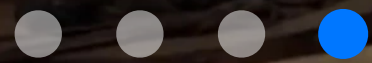
- Правка build.gradle(.kts)
- Отдельная ветка для ОК



Java 17, AGP 8

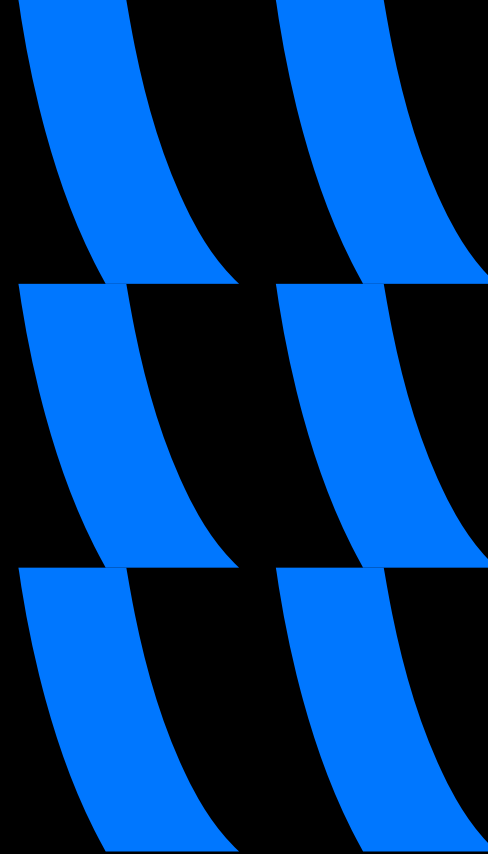
- Правка build.gradle(.kts)
- Отдельная ветка для ОК
- Переключение \$JAVA_HOME





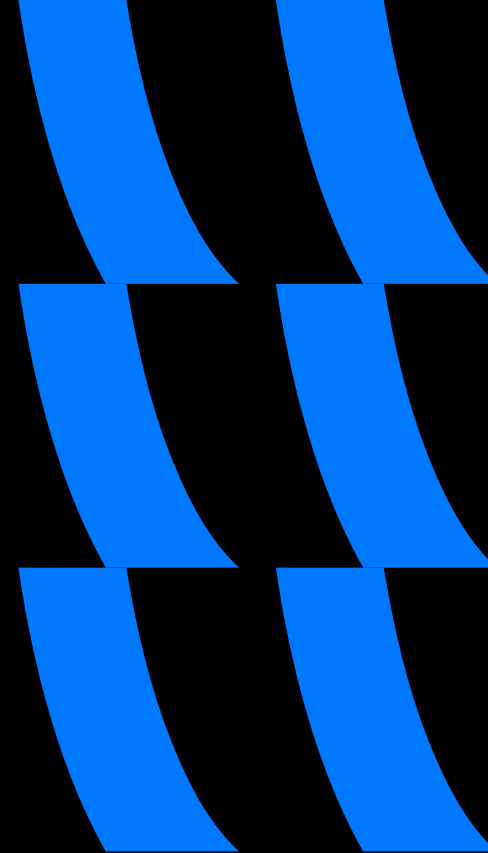
Стабилизация

Стабилизация



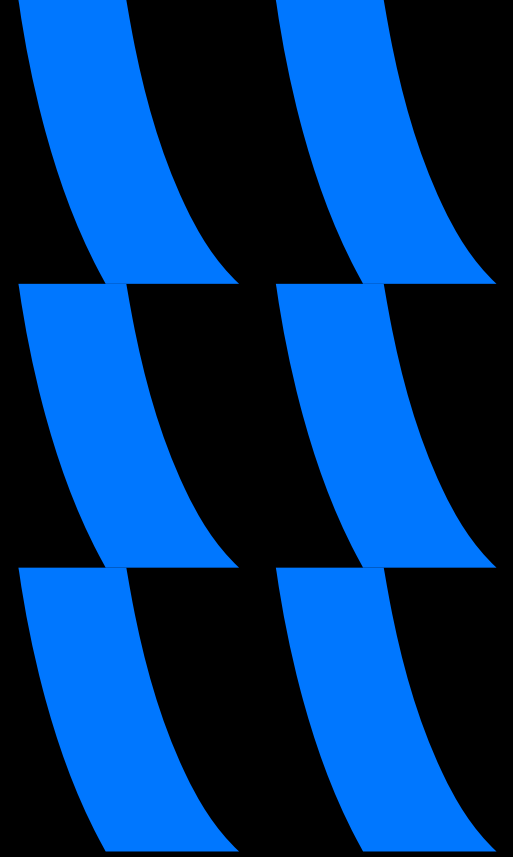
Стабилизация

- UI тесты



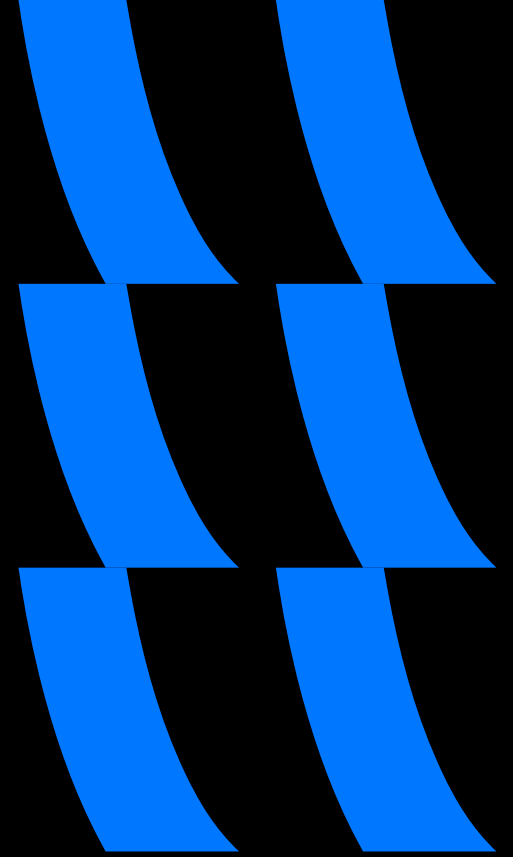
Стабилизация

- UI тесты
- Unit-тесты



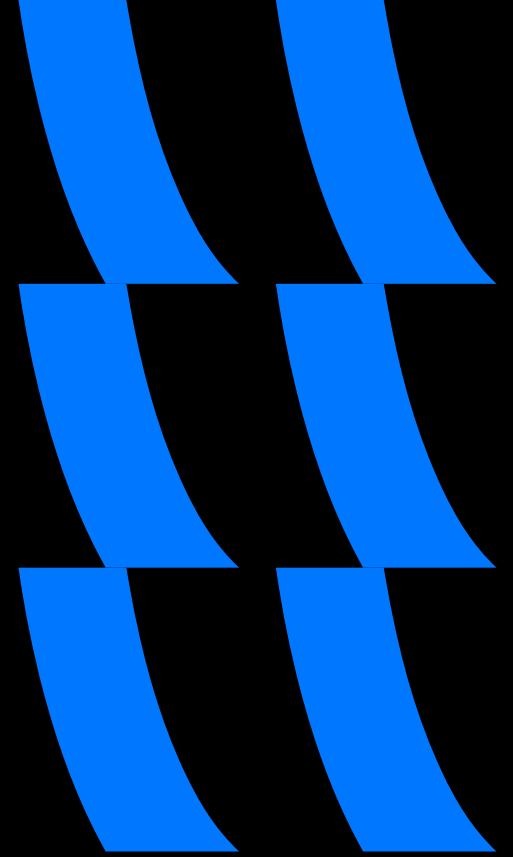
Стабилизация

- UI тесты
- Unit-тесты
- CI



Стабилизация

- UI тесты
- Unit-тесты
- CI
- Тестирование QA



Улучшайзинги

Улучшайзинги

Улучшайзинги

- Настроен includeBuild + mainframer для ОК

Улучшайзинги

- Настроен includeBuild + mainframer для ОК
- Пользователь всегда есть!

Улучшайзинги

- Настроен includeBuild + mainframer для ОК
- Пользователь всегда есть!
- Делегаты для управления функциями звонка

Улучшайзинги

- Настроен includeBuild + mainframer для ОК
- Пользователь всегда есть!
- Делегаты для управления функциями звонка
- Использование Dagger2, ViewInjection

Улучшайзинги

- Настроен includeBuild + mainframer для ОК
- Пользователь всегда есть!
- Делегаты для управления функциями звонка
- Использование Dagger2, ViewInjection

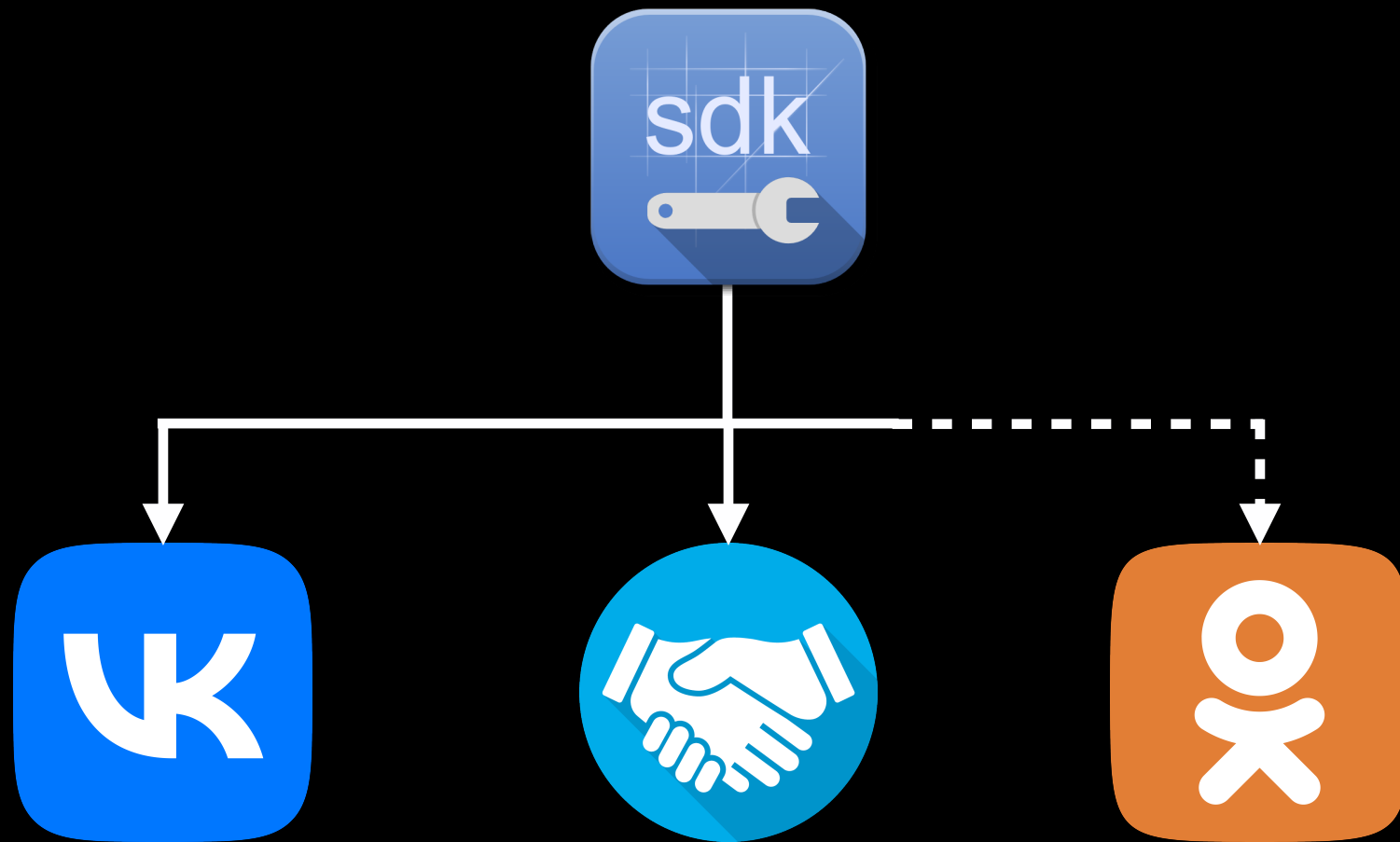




Как делать merge на 2 проекта?

Пользователи SDK

Пользователи SDK



Как делать merge на 2 проекта?

Как делать merge на 2 проекта?

1. Тестируем ОК + правки в SDK / исправление багов

Как делать merge на 2 проекта?

1. Тестируем ОК + правки в SDK / исправление багов
2. После тестирования – сборка SDK для ОК

Как делать merge на 2 проекта?

1. Тестируем ОК + правки в SDK / исправление багов
2. После тестирования – сборка SDK для ОК
3. Merge ОК с новыми зависимостями

Как делать merge на 2 проекта?

1. Тестируем ОК + правки в SDK / исправление багов
2. После тестирования – сборка SDK для ОК
3. Merge ОК с новыми зависимостями
4. Тестирование ВКонтакте + правки в SDK / исправление багов

Как делать merge на 2 проекта?

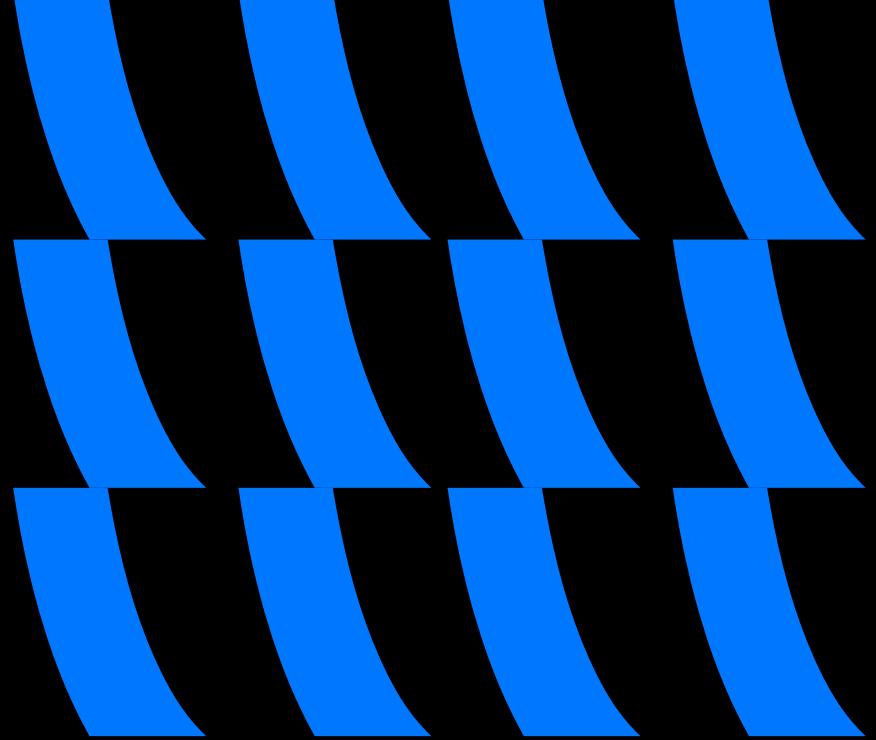
1. Тестируем ОК + правки в SDK / исправление багов
2. После тестирования – сборка SDK для ОК
3. Merge ОК с новыми зависимостями
4. Тестирование ВКонтакте + правки в SDK / исправление багов
5. Merge SDK и сборка стабильной версии

Как делать merge на 2 проекта?

1. Тестируем ОК + правки в SDK / исправление багов
2. После тестирования – сборка SDK для ОК
3. Merge ОК с новыми зависимостями
4. Тестирование ВКонтакте + правки в SDK / исправление багов
5. Merge SDK и сборка стабильной версии
6. Merge ВКонтакте со стабильной версией

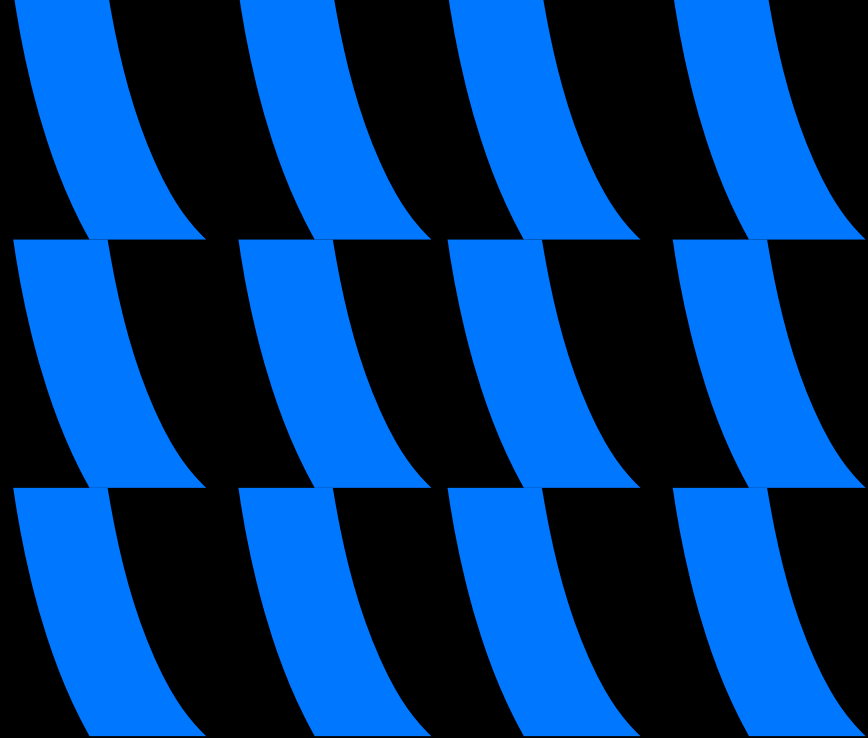
ИТОГИ

ИТОГИ



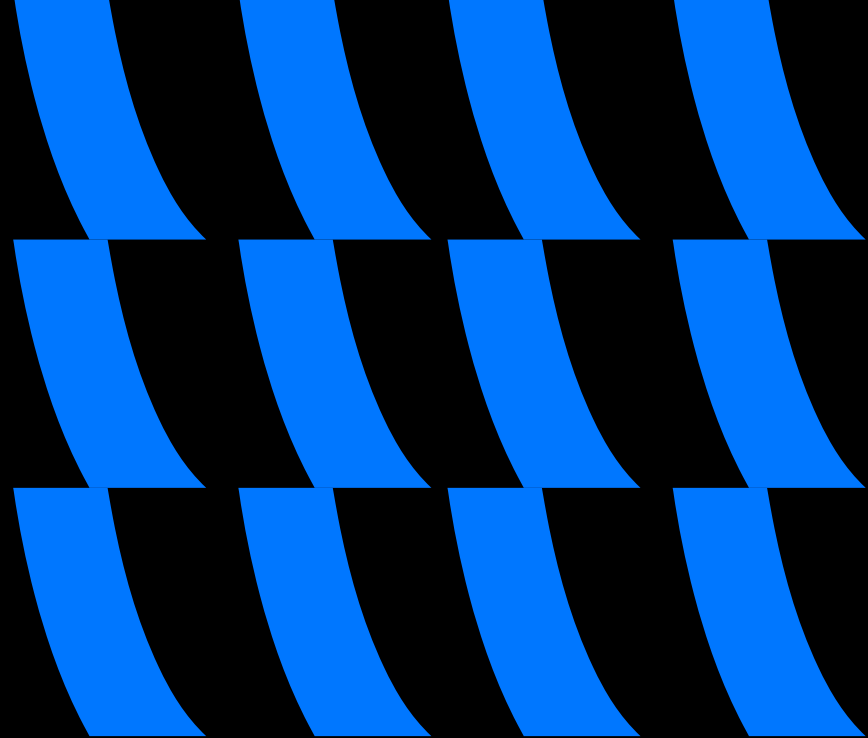
ИТОГИ

- Domain слой + DI



ИТОГИ

- Domain слой + DI
- UI фиксы



ИТОГИ

- Domain слой + DI
- UI фиксы



vladbot4 vladbot4

8 May at 16:23



Итоги

- Domain слой + DI
- UI фиксы
- Актуальная версия SDK и зависимостей



vladbot4 vladbot4

8 May at 16:23



ИТОГИ

- Domain слой + DI
- UI фиксы
- Актуальная версия SDK и зависимостей
- Исправлены баги в SDK и ОК



vladbot4 vladbot4

8 May at 16:23



ИТОГИ

- Domain слой + DI
- UI фиксы
- Актуальная версия SDK и зависимостей
- Исправлены баги в SDK и ОК
- Снята блокировка других платформ



vladbot4 vladbot4

8 May at 16:23



ИТОГИ

- Domain слой + DI
- UI фиксы
- Актуальная версия SDK и зависимостей
- Исправлены баги в SDK и ОК
- Снята блокировка других платформ
- Инфраструктура для разработки



vladbot4 vladbot4

8 May at 16:23



Этапы решения задачи

Этапы решения задачи

Знакомство с проектом и изоляция кода

Этапы решения задачи



Знакомство с проектом и изоляция кода

Этапы решения задачи



Знакомство с проектом и изоляция кода

Удаление старого движка

Этапы решения задачи



Знакомство с проектом и изоляция кода



Удаление старого движка

Этапы решения задачи



Знакомство с проектом и изоляция кода



Удаление старого движка

Подключение нового движка

Этапы решения задачи



Знакомство с проектом и изоляция кода



Удаление старого движка



Подключение нового движка

Этапы решения задачи



Знакомство с проектом и изоляция кода



Удаление старого движка



Подключение нового движка

Тестирование и стабилизация

Этапы решения задачи



Знакомство с проектом и изоляция кода



Удаление старого движка



Подключение нового движка



Тестирование и стабилизация

Этапы решения задачи



Знакомство с проектом и изоляция кода



Удаление старого движка



Подключение нового движка

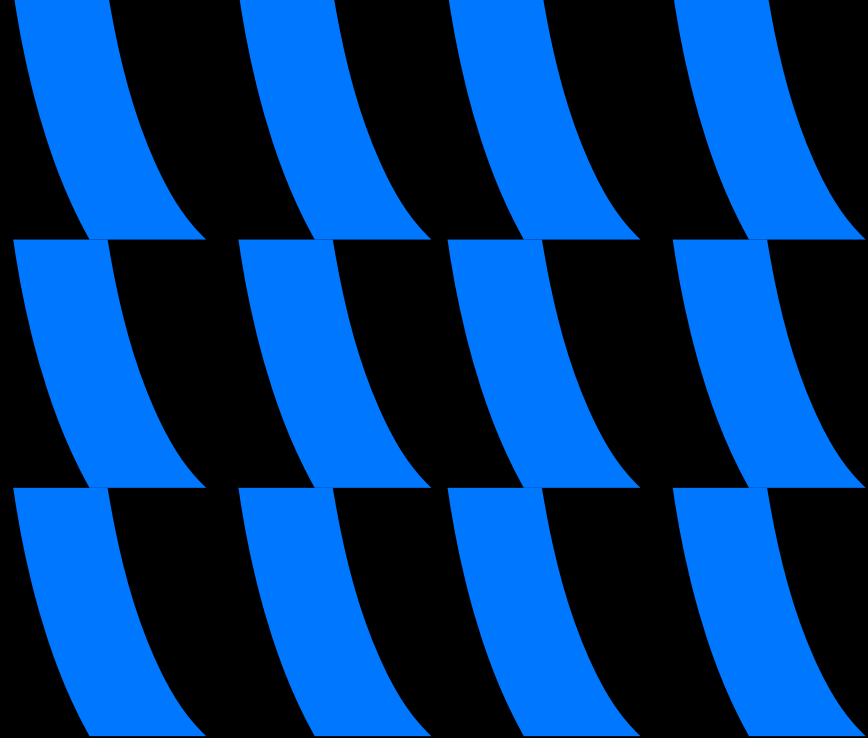


Тестирование и стабилизация

Раскатка на пользователей

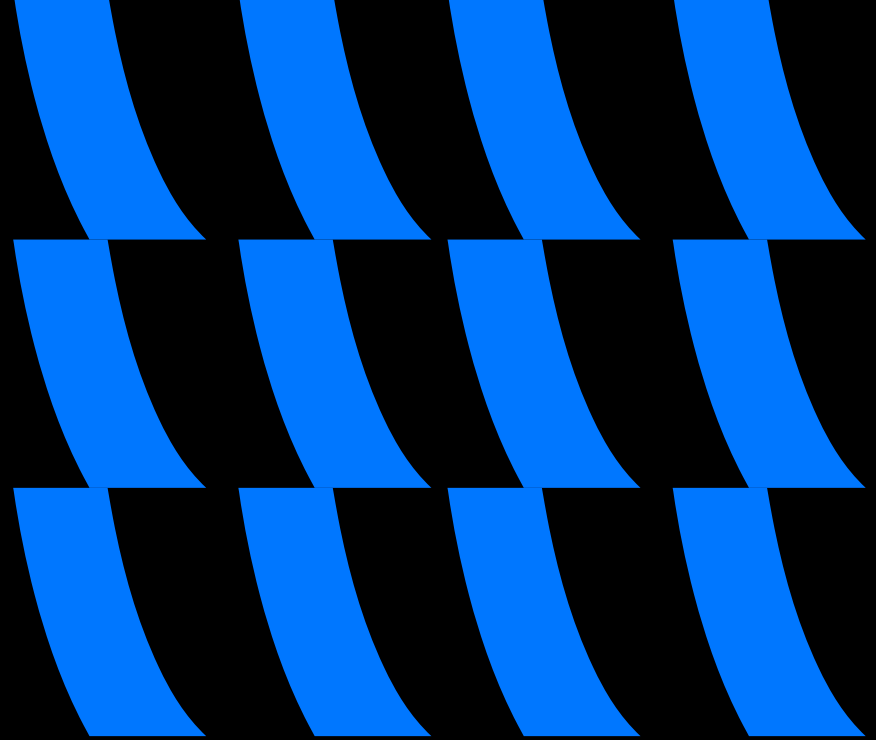
Рекомендации

Рекомендации



Рекомендации

- Иногда проще переписать код



Рекомендации

- Иногда проще переписать код



Рекомендации

- Иногда проще переписать код
- Редизайн + воскрешение



Рекомендации

- Иногда проще переписать код
- Редизайн + воскрешение
- Актуализация подзадач



Рекомендации

- Иногда проще переписать код
- Редизайн + воскрешение
- Актуализация подзадач
- Актуализируйте версии используемых библиотек



Рекомендации

- Иногда проще переписать код
- Редизайн + воскрешение
- Актуализация подзадач
- Актуализируйте версии используемых библиотек
- Пушкин SOLID, KISS, паттерны – наше все



Рекомендации

- Иногда проще переписать код
- Редизайн + воскрешение
- Актуализация подзадач
- Актуализируйте версии используемых библиотек
- Пушкин SOLID, KISS, паттерны – наше все
- Паттерны гибкости (Builder, Adapter, Strategy)



Рекомендации

- Иногда проще переписать код
- Редизайн + воскрешение
- Актуализация подзадач
- Актуализируйте версии используемых библиотек
- Пушкин SOLID, KISS, паттерны – наше все
- Паттерны гибкости (Builder, Adapter, Strategy)
- API и Impl модули, маппинг моделей

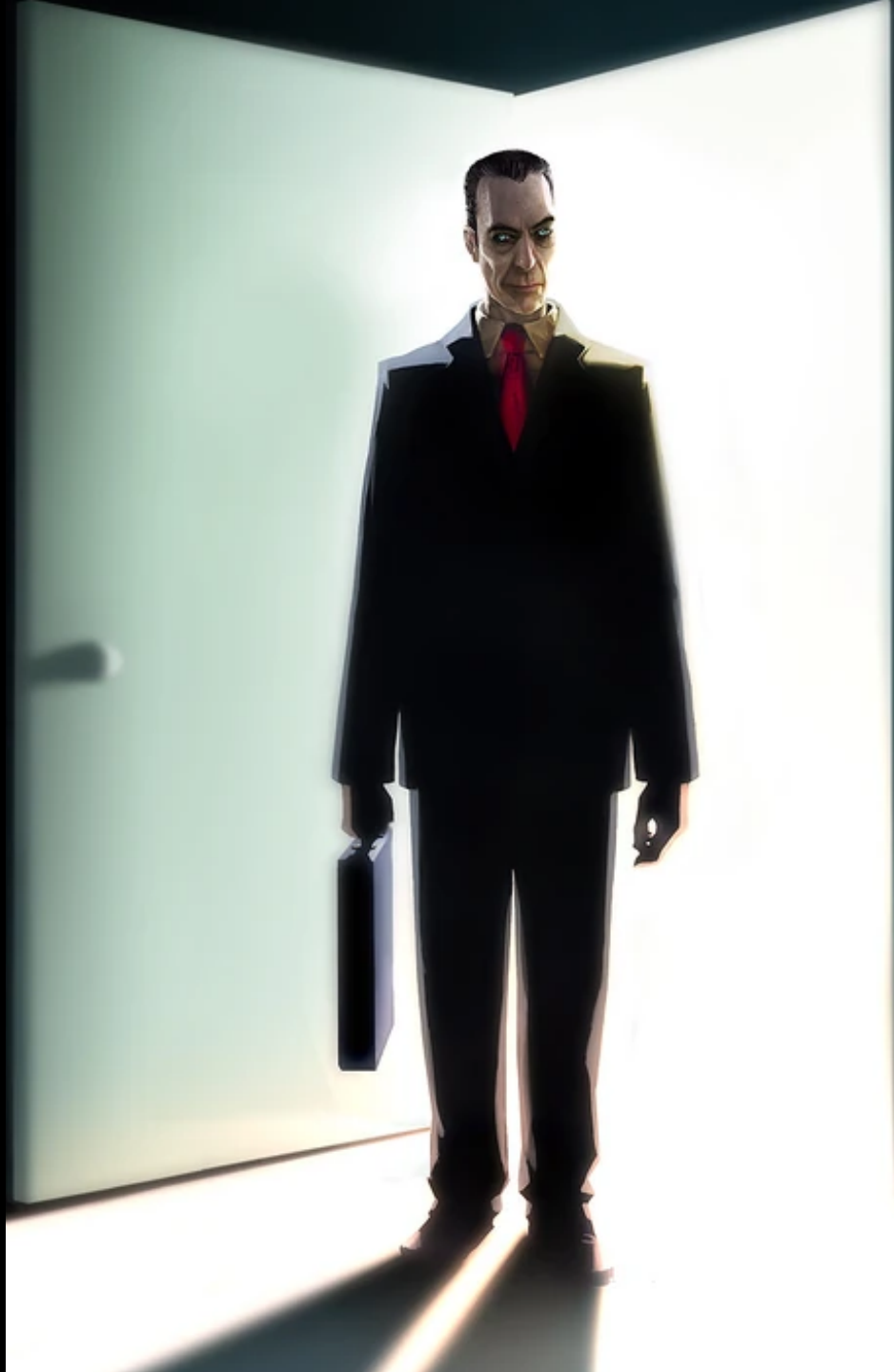


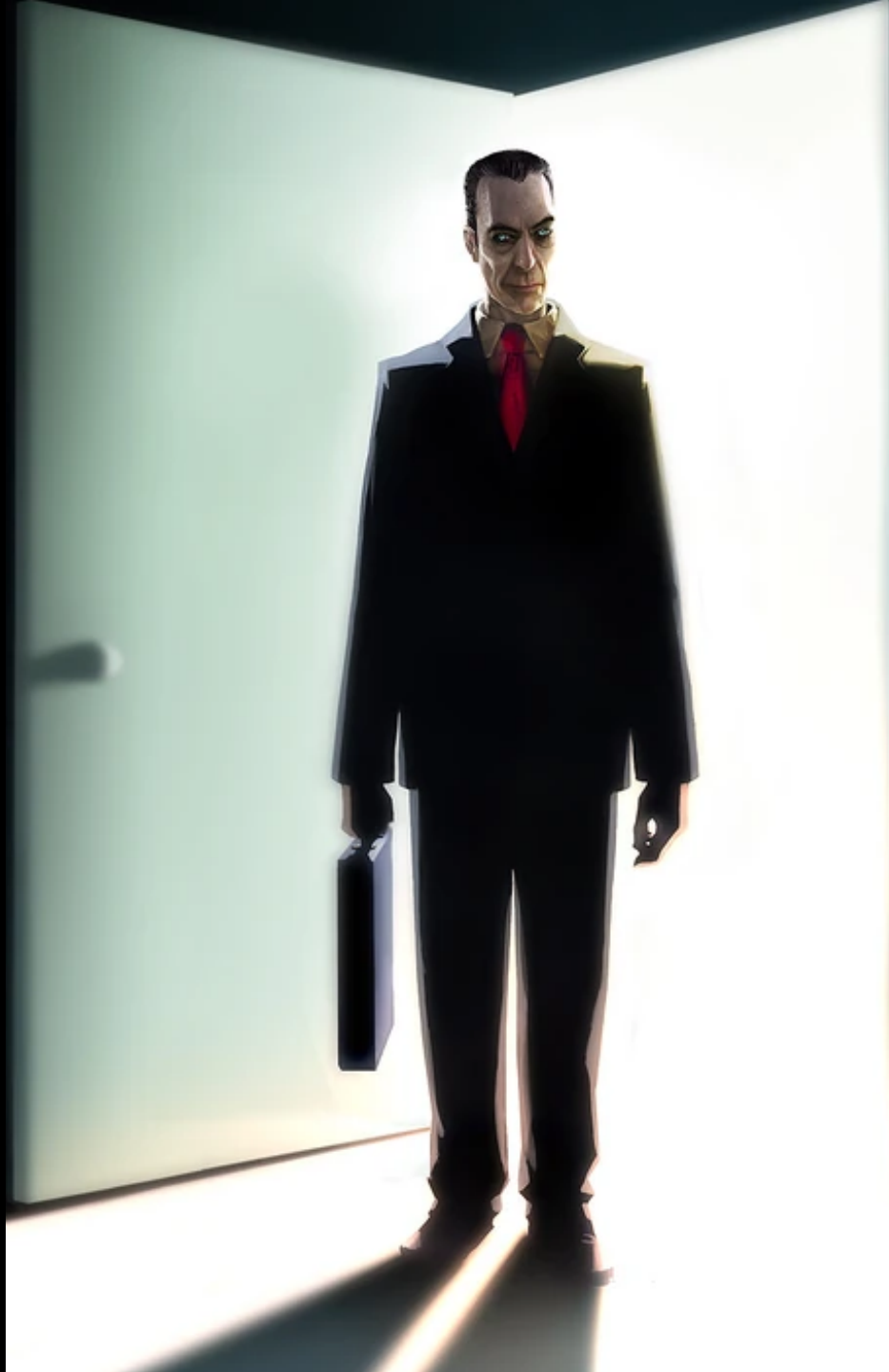
Рекомендации

- Иногда проще переписать код
- Редизайн + воскрешение
- Актуализация подзадач
- Актуализируйте версии используемых библиотек
- Пушкин SOLID, KISS, паттерны – наше все
- Паттерны гибкости (Builder, Adapter, Strategy)
- API и Impl модули, маппинг моделей
- Отдых и хобби









Проект ушел
в стазис

Будем ВКонтакте!

Владислав Кожушко

v.kozhushko@vk.team

