

📍 @IT\_kmr

📍 @sborkacamp



Сборка\_



**GoodLine**  
Оператор связи Кузбасса



АТВИНТА



Сборка\_

# IT-COMMUNITY KEMEROVO



[t.me/IT\\_kmr](https://t.me/IT_kmr)

Вступай, общайся, спрашивай, отвечай...



**GoodLine**  
Оператор связи Кузбасса



АТВИНТА



Сборка\_

???\*

\* - Вопросы наше все !

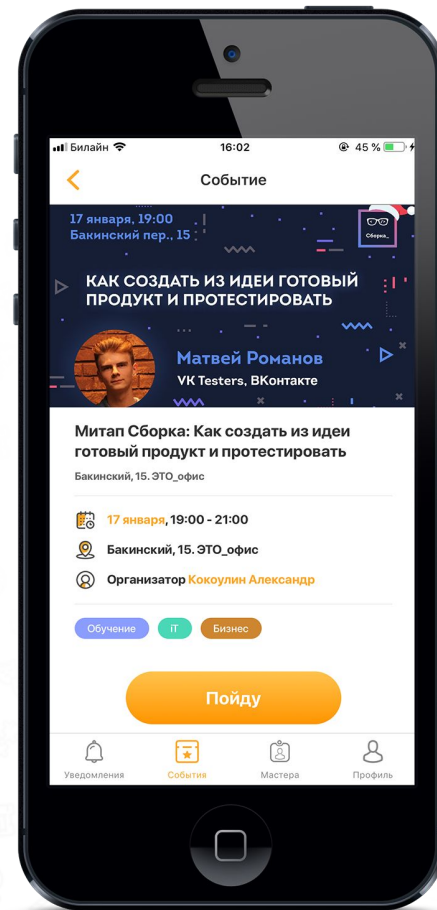
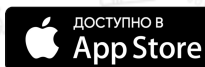


АТВИНТА

# Устанавливайте приложение



- можно нажать **Пойду** и получать напоминание о мероприятии
- обсуждать мероприятие
- делиться мероприятием с друзьями



# Навигация в Android

от UX до реализации

Севастьян Жуков

Android Developer

 @seva\_zhukov



Новые фи́чи



Усложнение навигации



Удобный UX

Навигация — процесс управления некоторым объектом

Навигация в Android — перемещение между фичами



Панель навигации  
+  
Программная реализация  
=  
Навигация

OT UX

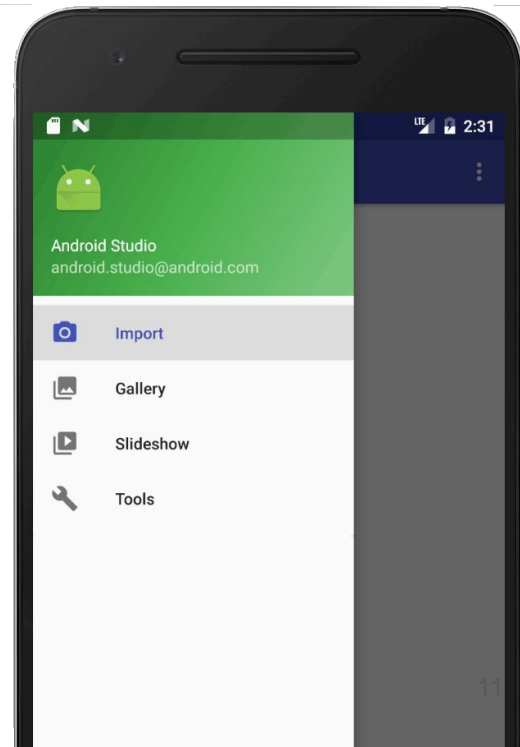
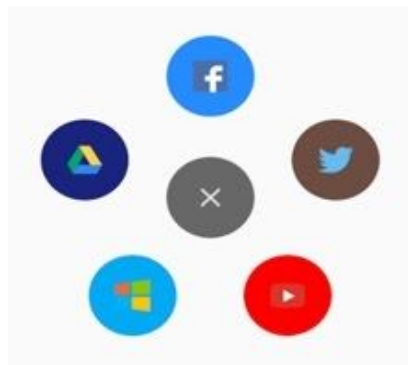
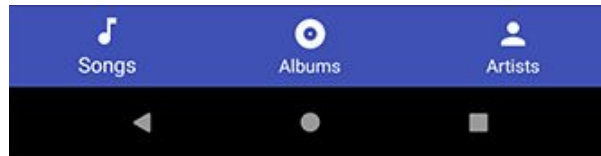
Drawer Layout

Tab Navigation

Circle Menu

Classic Menu

Гибридные решения

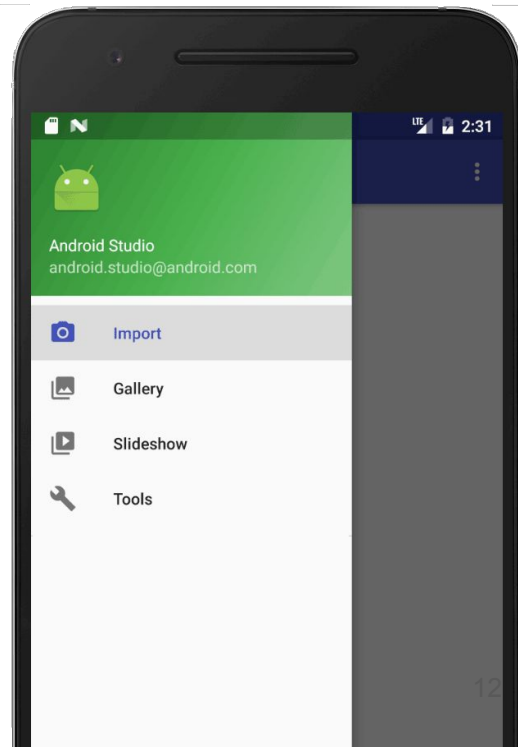


# Drawer Layout

— выдвигающаяся панель сбоку

Доступ к фиче за 2 действия

В приложении предполагается большое количество важных фич (более 5) и доступ к ним хотелось бы иметь за минимальное количество шагов



# Drawer Layout

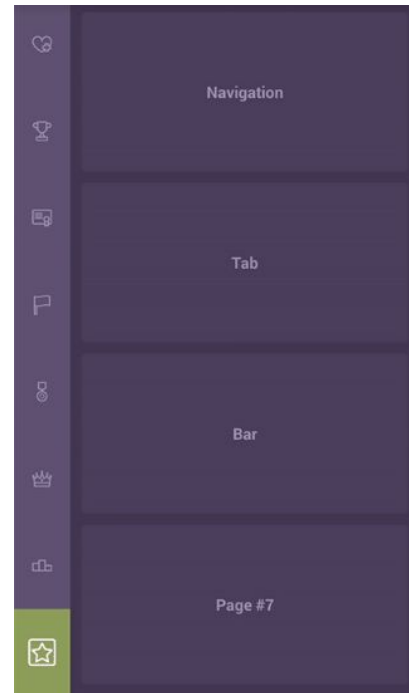
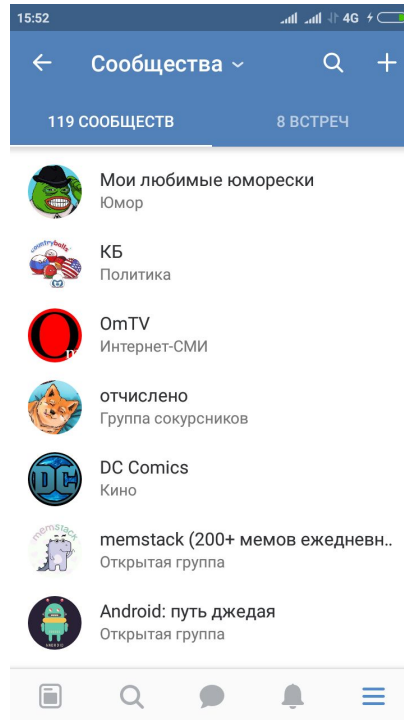
- + Много фич на панели
- + Привычно для пользователя
- Сложная настройка
- Сложности доступа на большом экране (тяжело тянуться пальцем в противоположную сторону экрана)

# Tab Navigation

— панель внизу экрана

Доступ к фиче за 1 действие

Небольшое количество  
главных фич (менее 5)



# Tab Navigation

- + Доступ к фице в один клик
- Возможность разместить на панели малое количество фич

# Circle Menu

— круглая кнопка в углу для отображения панели

Доступ к фиче за 2 тапа

Приложение развлекательного портала +  
большое количество важных фич (до 9)





# Circle Menu

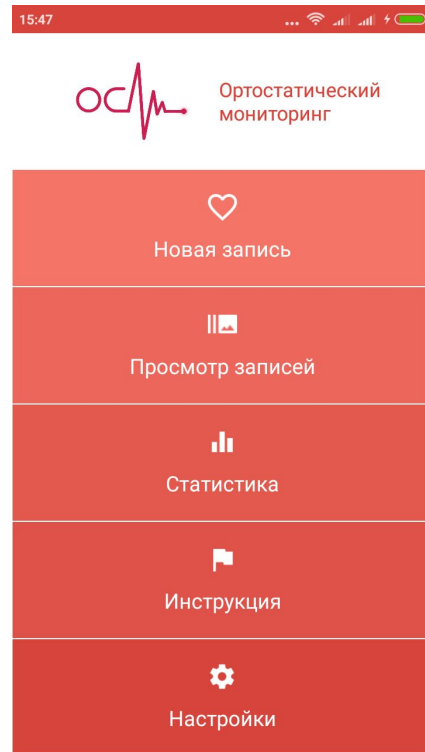
- + Ярко, оригинально и динамично
- + Большое количество фич (до 9)
- Нагрузка приложения анимацией
- Кнопка занимает часть контента
- Непривычно пользователю

# Classic Menu

— панель занимает весь экран и  
содержит в себе все фичи  
приложения

Доступ к фиче за 1 тап

Небольшое приложение,  
просто для новичков



# Classic Menu

- + Простота реализации
- + Очевидность использования
- Невозможность перейти к другой фиче, не вернувшись на главную панель навигации

# Гибридные решения

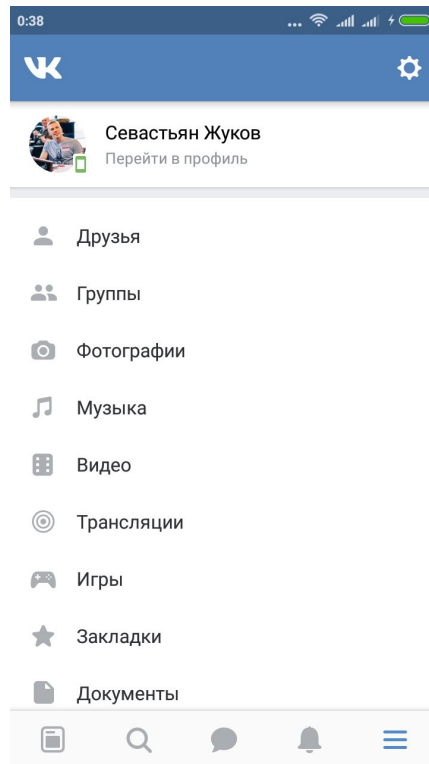
Невозможно использовать  
только один вид панели

ВК: много важных фич

Tab + Classic

Lingualeo: 2 логических  
раздела в одной из фич

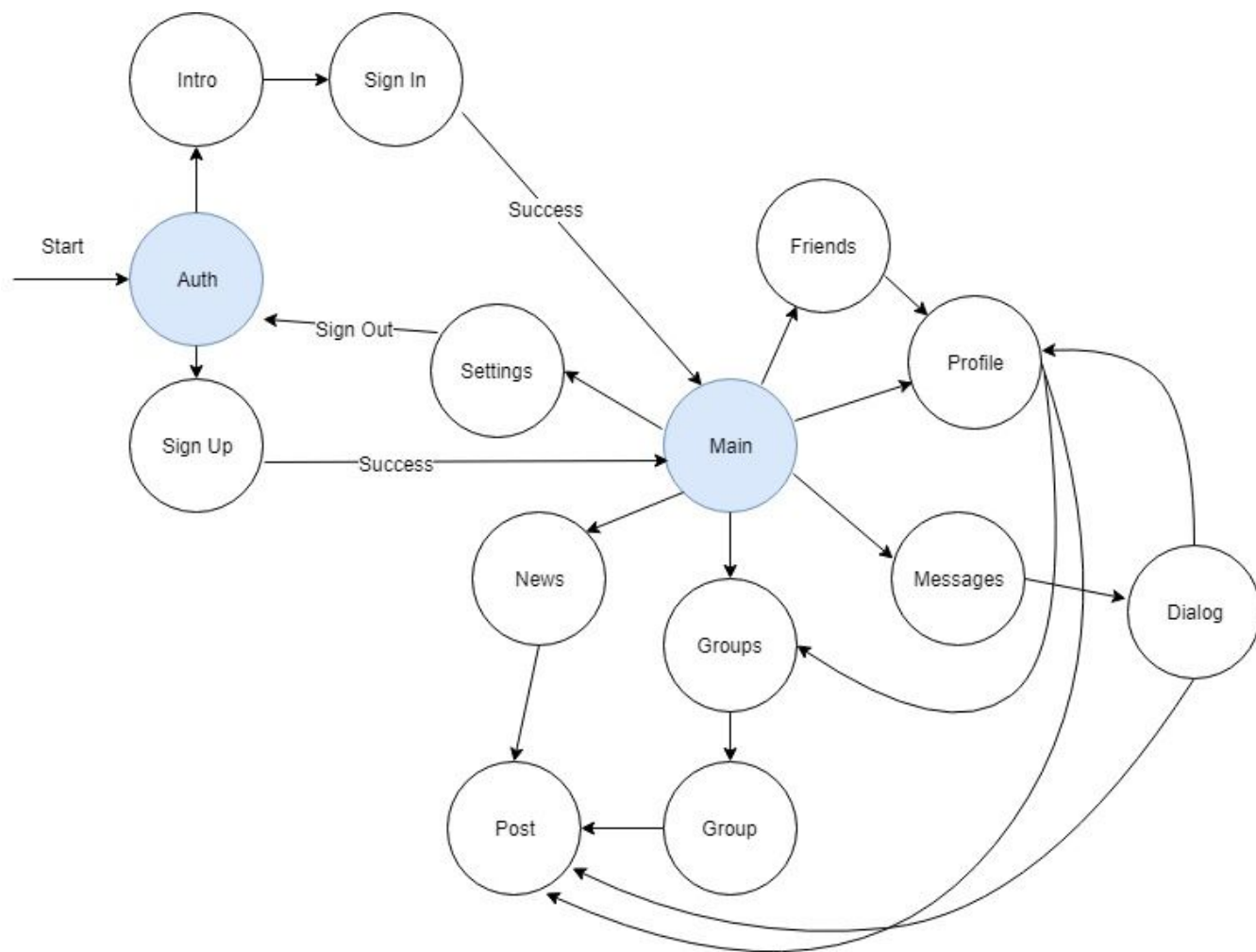
Drawer + Tab



# Выбираем решение

Вид\Параметр	Более 5 главных фич	Неопытность разработчика	Очевидность интерфейса	Оригинальнос ть интерфейса	Возможность большой вложенности фич	Минимальный шаг доступа к фиче
Drawer Layout	+2	0	+2	+1	+2	+1
Tab Navigation	0	+1	+2	+1	+2	+2
Circle Menu	+1	+1	0	+2	+2	+1
Classic Menu	+1	+2	+2	0	0	0

до реализации



Set Root

Push

Replace

Clear stack + Add New Screen

Back / BackTo

Change container



Cicerone

Conductor

Jetpack

MultiStack Navigation  
Library

Решение

# FragmentManager

## Set Root

```
fragmentManager.beginTransaction()  
    .add(R.id.container, IntroFragment())  
    .commit()
```

## Replace

```
fragmentManager.beginTransaction()  
    .replace(R.id.container, SignInFragment())  
    .commit()
```

# FragmentManager

Push = Replace + AddToBackStack

```
FragmentManager.beginTransaction()  
    .replace(R.id.container, SignUpFragment())  
    .addToBackStack(Screen.SIGN_UP)  
    .commit()
```

BackTo

```
supportFragmentManager.popBackStack(Screen.FIRST.name,  
FragmentManager.POP_BACK_STACK_INCLUSIVE)
```

Back

```
supportFragmentManager.popBackStack()
```

# FragmentManager

## Clear stack + Set New Screen

```
for (i in 0 until fragmentManager.backStackEntryCount) {  
    fragmentManager.popBackStackImmediate()  
}  
fragmentManager  
    .beginTransaction()  
    .replace(R.id.container, fragment)  
    .commit()
```

# FragmentManager

Переход между контейнерами (активити)

```
startActivity(Intent(this, MainActivity::class.java))
```

Передача данных между фрагментами

```
val bundle = Bundle()  
bundle.putString("id", "1")  
val postFragment = PostFragment()  
postFragment.arguments = bundle
```

# FragmentManager.Проблемы

Много кода

Нет сохранения состояния навигатора

Нужно учитывать жизненный цикл контейнера

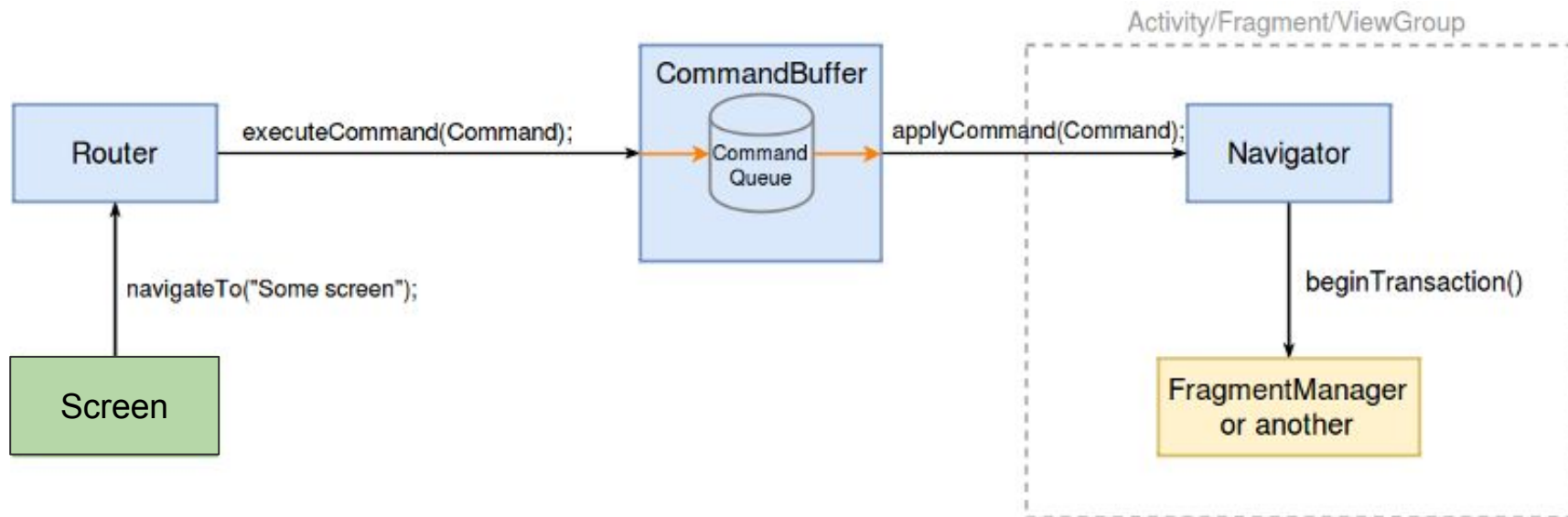
**java.lang.IllegalStateException: Can not perform  
this action after onSaveInstanceState**

# FragmentManager.Решение проблем

## Extensions

```
fun Fragment.setClearScreen(containerId: Int, fragment: Fragment, ) {  
    for (i in 0 until fragmentManager!!.backStackEntryCount) {  
        fragmentManager!!.popBackStackImmediate()  
    }  
    fragmentManager!!  
        .beginTransaction()  
        .replace(containerId, fragment)  
        .commit()  
}
```

# Cicerone





# Cicerone

```
cicerone = Cicerone.create()
```

```
fun getNavigatorHolder(): NavigatorHolder {  
    return cicerone.navigatorHolder  
}
```

```
fun getRouter(): Router {  
    return cicerone.router  
}
```

```
var navigator = object :  
    SupportFragmentManager(supportFragmentManager,  
        R.id.main_container) {  
        override fun createFragment(screenKey: String?, data:  
Any?): Fragment {  
            return when (screenKey) {  
                Screen.FIRST.name -> FirstScreen() ...  
            } ...  
        }  
    }  
override fun onResume() {  
    super.onResume()  
    App.instance.getNavigatorHolder().setNavigator(navigator)  
}
```

# Cicerone

## Set Root

```
getRouter().newRootScreen(Screen.FIRST.name)
```

## Replace

```
getRouter().replaceScreen(Screen.SECOND.name)
```

## Push

```
getRouter().navigateTo(Screen.SECOND.name)
```

# Cicerone

Clear stack + Set New Screen

```
getRouter().newRootScreen(Screen.FIRST.name)
```

Передача данных между экранами

```
getRouter().navigateTo(Screen.SECOND.name, data)
```

BackTo

```
App.instance.getRouter().backTo(Screen.FIRST.name)
```

# Cicerone

Back

```
App.instance.getRouter().exit()
```

Переход между контейнерами (активити)

```
getGlobalRouter().navigateTo(Screen.MAIN.name)
```

# Cicerone

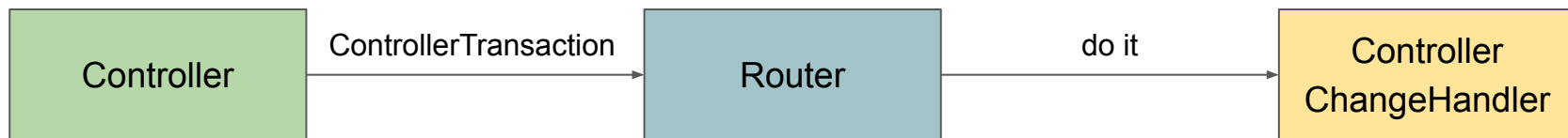
Сохранение состояния навигатора

Транзакции в одну строку

Экраны - не обязательно фрагменты

Дополнительный код реализации навигации

# Conductor



# Conductor

## Router

```
val router = Conductor.attachRouter(this, auth_container,  
savedInstanceState)
```

## Set Root

```
router.setRoot(RouterTransaction.with(IntroController()))
```

## Replace

```
router.replaceTopController(RouterTransaction.with(SignInCon  
troller()))
```



# Conductor

## Push

```
router.pushController(RouterTransaction.with(SignUpController()))
```

## Clear stack + Set New Screen

```
router.setRoot(RouterTransaction.with(IntroController()))
```

## Back

```
router.handleBack()
```

# Conductor

Передача данных между экранами

```
router.pushController(RouterTransaction.with(SignUpController(data)))
```

BackTo

```
router.popToTag(TAG)
```

Переход между контейнерами

```
startActivity(Intent(this, MainActivity::class.java))
```

# Conductor.Интеграция с архитектурой

Mosby (MVP)

Controller - MvpController

Presenter создается после onCreateView и до onAttach

```
override fun onAttach(view: View) {  
    super.onAttach(view)  
    presenter.doSomething()  
}
```

# Conductor

Сохранение состояния навигатора

Транзакции в одну строку

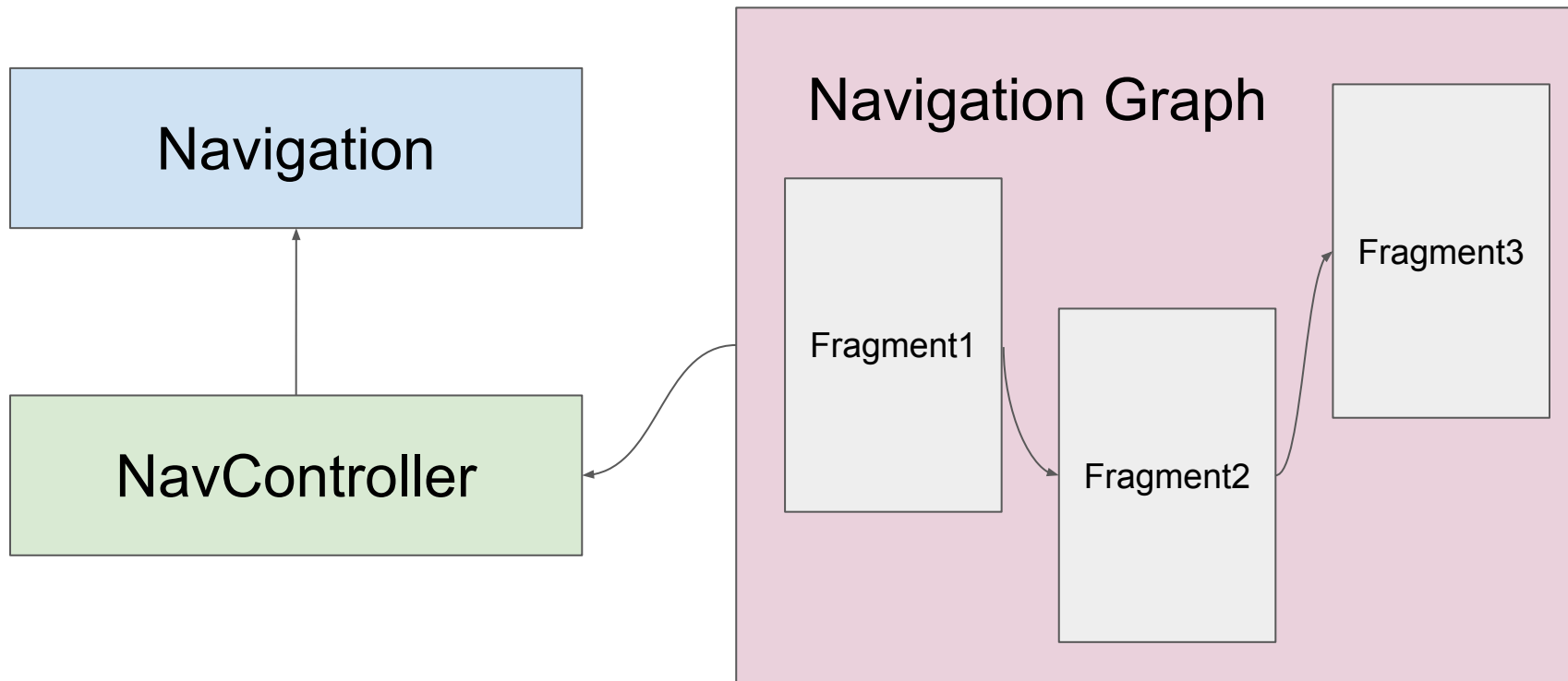
Быстрые транзакции

Легко интегрируемая анимация

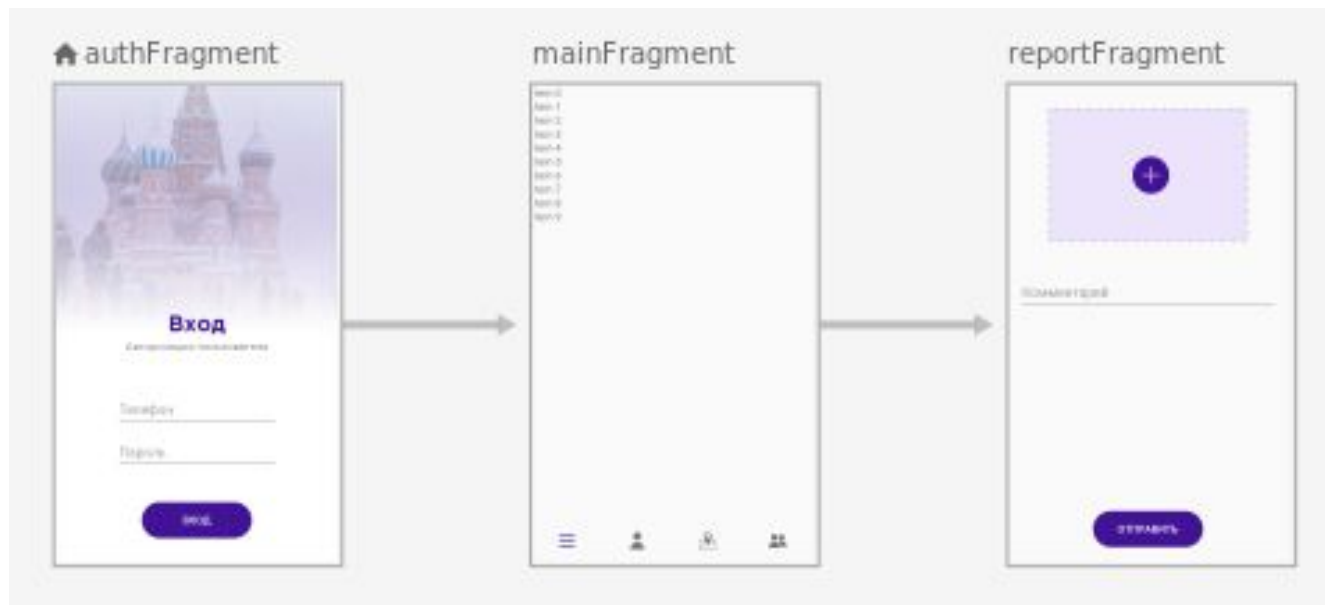
Не нужна “ручная” реализация навигации

Ограниченный выбор готовых архитектурных решений

# Navigation Architecture Component



# Navigation Architecture Component



HOST —————

activity\_main (nav\_host\_glo

GRAPH —————

- authFragment - Start
- mainFragment
- reportFragment

<navigation

...

```
    android:id="@+id/mobile_navigation"  
    app:startDestination="@id/authFragment">
```

<fragment

```
    android:id="@+id/authFragment"  
    android:name="com.memebattle.flexible_control.feature.auth.presentation.AuthFrag  
ment"
```

```
    android:label="fragment_auth"  
    tools:layout="@layout/fragment_auth" >
```

<action

```
    android:id="@+id/action_authFragment_to_mainFragment"  
    app:clearTask="true"  
    app:destination="@id/mainFragment" />
```

</fragment>

...

</navigation>

# Navigation Architecture Component

```
class MainActivity : AppCompatActivity() {
```

```
    lateinit var navController: NavController
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
```

```
        super.onCreate(savedInstanceState)
```

```
        setContentView(R.layout.activity_main)
```

```
        App.component.inject(this)
```

```
        navController = Navigation.findNavController(this,  
R.id.nav_host_global)
```

```
        navController.navigate(R.id.action_authFragment_to_mainFragment)
```

```
    }  
}
```



# Navigation Architecture Component

Set Root

in Navigation Graph: `app:startDestination="@id/authFragment"`

Clear stack + Set New Screen

in action: `app:clearTask="true"`

Передача данных между экранами

`navController.navigate(R.id.mapFragment, args)`

# Navigation Architecture Component

Push

```
navController.navigate(R.id.action_authFragment_to_mainFragment)
```

Back

```
navController.popBackStack()
```

BackTo

```
navController.popBackStack(R.id.mapFragment, true)
```

# Deep Link

Запуск приложения сразу на нужном экране

Установка Deep Link в destination

```
<deepLink
```

```
    android:id="@+id/deepLink"
```

```
    app:uri="app://myapp/frag3" />
```

# Deep Link

`AndroidManifest.xml`

```
<activity  
    android:name=".MainActivity">  
    <nav-graph android:value="@navigation/main_graph" />  
</activity>
```



Авто-настройка Intent Filter

# Navigation Architecture Components

## Navigation UI

Автоматическое переключение по стекам

`NavigationUI.setupWithNavController(bottomNavigationView, navController)`

**Стеки экранов не сохраняются!**

# Navigation Architecture Component

Наглядный граф навигации

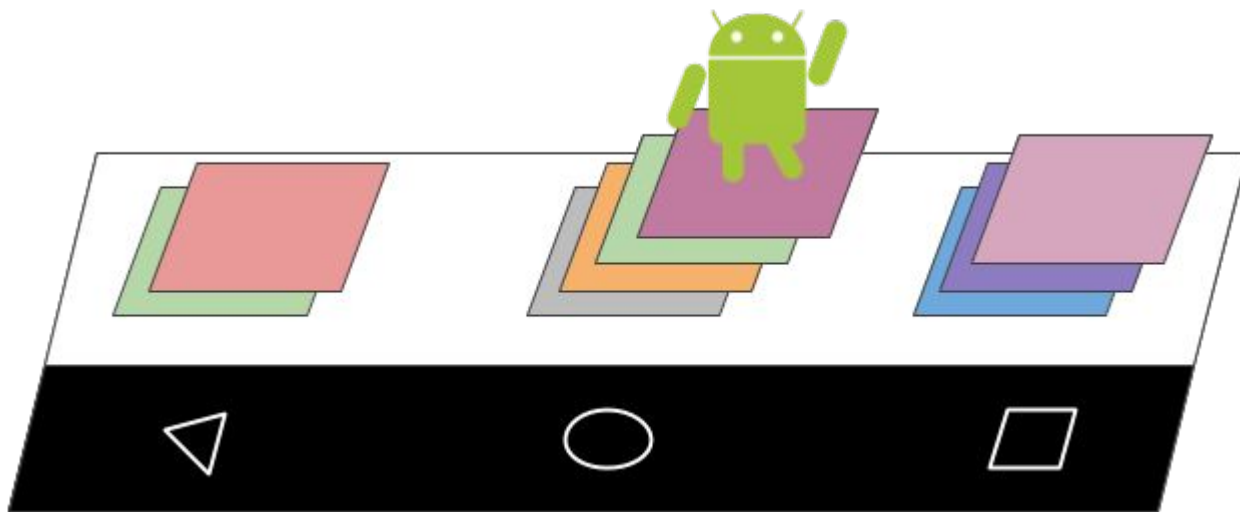
Deep Link

Nested Graph

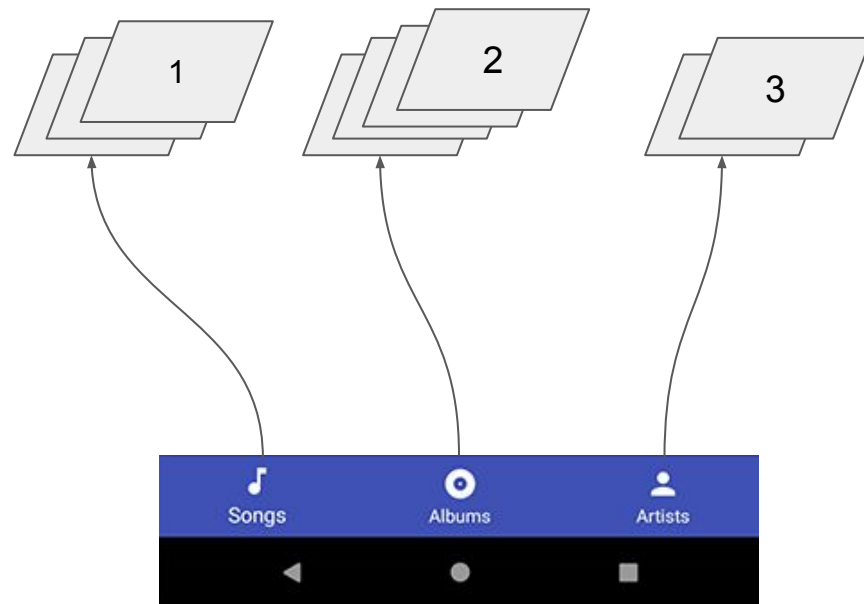
На графе нельзя описать переходы назад

`java.lang.IllegalStateException: Can not perform this action after onSaveInstanceState`

# Заряжаем колоды фрагментов с MultiStack Navigation Library



# СТЭКИ





# Порядок стеков

tap **profile** item [**profile**]

tap **news** item [**profile**, **news**]

tap **messages** item [**profile**, **news**, **messages**]

tap **news** item [**profile**, **messages**, **news**]

# Pop Screen

Спуск по стеку `item` а

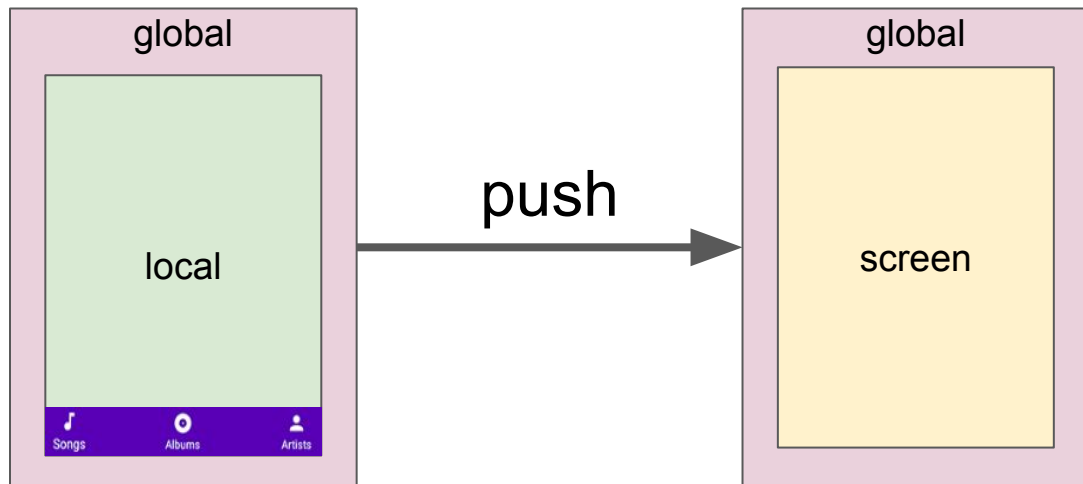
Если стек `item` а заканчивается, то происходит спуск по порядку стеков

`[profile, news]`

`[profile]`

# Глобальный Push Screen

Есть возможность сделать **Push Screen** в глобальный контейнер, оставаясь в локальном бэкстеке



# Настройка MSNL.Activity

```
class MainActivity : AppCompatActivity() {  
    private lateinit var msFragmentManager: MSFragmentManager  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        msFragmentManager = MSFragmentManager(supportFragmentManager)  
        msFragmentManager.globalContainerId = R.id.global_container  
        msFragmentManager.addGlobal(MainFragment())  
    }  
}
```

# Настройка MSNL.Activity

```
class MainActivity : AppCompatActivity() {  
    private lateinit var msFragmentManager: MSFragmentManager  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        msFragmentManager = MSFragmentManager(supportFragmentManager)  
        msFragmentManager.globalContainerId = R.id.global_container  
        msFragmentManager.addGlobal(MainFragment())  
    }  
}
```

# Настройка MSNL.Activity

```
class MainActivity : AppCompatActivity() {  
    private lateinit var msFragmentManager: MSFragmentManager  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        msFragmentManager = MSFragmentManager(supportFragmentManager)  
        msFragmentManager.globalContainerId = R.id.global_container  
        msFragmentManager.addGlobal(MainFragment())  
    }  
}
```

# Настройка MSNL.Activity

```
class MainActivity : AppCompatActivity() {  
    private lateinit var msFragmentManager: MSFragmentManager  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        msFragmentManager = MSFragmentManager(supportFragmentManager)  
        msFragmentManager.globalContainerId = R.id.global_container  
        msFragmentManager.addGlobal(MainFragment())  
    }  
}
```

# Настройка MSNL.Activity

```
class MainActivity : AppCompatActivity() {  
    private lateinit var msFragmentManager: MSFragmentManager  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        msFragmentManager = MSFragmentManager(supportFragmentManager)  
        msFragmentManager.globalContainerId = R.id.global_container  
        msFragmentManager.addGlobal(MainFragment())  
    }  
}
```



# Настройка MSNL.Activity

```
override fun onBackPressed() {  
    val fragment = this.supportFragmentManager  
        .findFragmentById(R.id.global_container)  
    (fragment as? IOnBackPressed)?.onBackPressed()  
}
```

# Настройка MSNL.FlowFragment

```
class MainFragment : Fragment(), IOnBackPressed {  
    override fun onCreateView(inflater: LayoutInflater, container:  
ViewGroup?, savedInstanceState: Bundle?): View? {  
        val v = inflater.inflate(R.layout.fragment_main, container, false)  
        msFragmentManager.localContainerId = R.id.local_container  
        val fragments = arrayListOf(NewsFragment(), MessagesFragment(),  
FriendsFragment(), ProfileFragment())  
        MSNavigation.setupNavigation(msFragmentManager,  
v.bottomNavigationView, fragments)  
        return v  
    }  
}
```

# Настройка MSNL.FlowFragment

```
class MainFragment : Fragment(), IOnBackPressed {  
    override fun onCreateView(inflater: LayoutInflater, container:  
ViewGroup?, savedInstanceState: Bundle?): View? {  
        val v = inflater.inflate(R.layout.fragment_main, container, false)  
        msFragmentManager.localContainerId = R.id.local_container  
        val fragments = arrayListOf(NewsFragment(), MessagesFragment(),  
FriendsFragment(), ProfileFragment())  
        MSNavigation.setupNavigation(msFragmentManager,  
v.bottomNavigationView, fragments)  
        return v  
    }  
}
```

# Настройка MSNL.FlowFragment

```
class MainFragment : Fragment(), IOnBackPressed {  
    override fun onCreateView(inflater: LayoutInflater, container:  
ViewGroup?, savedInstanceState: Bundle?): View? {  
        val v = inflater.inflate(R.layout.fragment_main, container, false)  
        msFragmentManager.localContainerId = R.id.local_container  
        val fragments = arrayListOf(NewsFragment(), MessagesFragment(),  
FriendsFragment(), ProfileFragment())  
        MSNavigation.setupNavigation(msFragmentManager,  
v.bottomNavigationView, fragments)  
        return v  
    }  
}
```

# Настройка MSNL.FlowFragment

```
class MainFragment : Fragment(), IOnBackPressed {  
    override fun onCreateView(inflater: LayoutInflater, container:  
ViewGroup?, savedInstanceState: Bundle?): View? {  
        val v = inflater.inflate(R.layout.fragment_main, container, false)  
        msFragmentManager.localContainerId = R.id.local_container  
        val fragments = arrayListOf(NewsFragment(), MessagesFragment(),  
FriendsFragment(), ProfileFragment())  
  
        MSNavigation.setupNavigation(msFragmentManager,  
v.bottomNavigationView, fragments)  
  
        return v  
    }  
}
```

# Настройка MSNL.FlowFragment

```
override fun onBackPressed(): Boolean {  
    MSNavigation.onBackPressed()  
    return true  
}
```

# MSNL.Методы

```
fun navigate(fragment: Fragment, args: Bundle?)
fun replace(fragment: Fragment, args: Bundle?)
fun navigateGlobal(fragment: Fragment, args: Bundle?)
fun replaceGlobal(fragment: Fragment, args: Bundle?)
fun add(containerId: Int, fragment: Fragment)
fun addGlobal(containerId: Int, fragment: Fragment)
fun back()
fun backTo(fragmentTag: String)
```

# Вопросы?



MSNL

Севастьян Жуков

 @seva\_zhukov





Сборка\_

# СЛЕДИТЕ ЗА НАМИ



[t.me/sborkacamp](https://t.me/sborkacamp)



[t.me/IT\\_kmr](https://t.me/IT_kmr)



[vk.com/sborkacamp](https://vk.com/sborkacamp)



АТВИНТА