

streaming  
services

# Single Activity by Example

Fragments are the new Activities

# Why this talk?

Single Activity by Example

**“Until they really fix Backstack Management  
with fragments this is still going to be painful.”**

**-Twitter User**

**“Don’t get the Single Activity hype.  
It doesn’t scale.”**

**-Twitter User**

**“Big projects should be split up in  
feature modules, one Activity per feature.”**

**-Twitter User**

**“Single activity makes it too easy to create spaghetti, that is likely going to be what you get. Multiple activities force you to decouple better.”**

**-Twitter User**

# Points to discuss

- Fragment Back Stack
- Single Activity Scalability

9:00



# Settings

Push Notifications



My Team



Scoreboard



Settings

# Fragment Navigation

Back stack issues?

# What is a backstock?

# Stack of cards

Only the top franchise hits have a View

All Fragments in the stack are CREATED

```
1 fragmentManager.commit {  
2     replace(R.id.main_activity_container, FragmentA())  
3 }  
4 fragmentManager.commit {  
5     replace(R.id.main_activity_container, FragmentB())  
6     addToBackStack(null)  
7 }
```

~~DESIGNATED~~

~~RECEIVED~~

FragmentB

FragmentA

on ~~Desktop View~~

on ~~Desktop View~~

# Unnecessary view creation

# setReorderingAllowed

```
1 fragmentManager.commit {  
2     replace(R.id.main_activity_container, FragmentA())  
3     setReorderingAllowed(true)  
4 }  
5 fragmentManager.commit {  
6     replace(R.id.main_activity_container, FragmentB())  
7     setReorderingAllowed(true)  
8     addToBackStack(null)  
9 }
```

## First Transaction

onResume

onStart

onCreateView

onCreate

## Second Transaction

onResume

onStart

onCreateView

onCreate

~~REDACTED~~

ScreenB

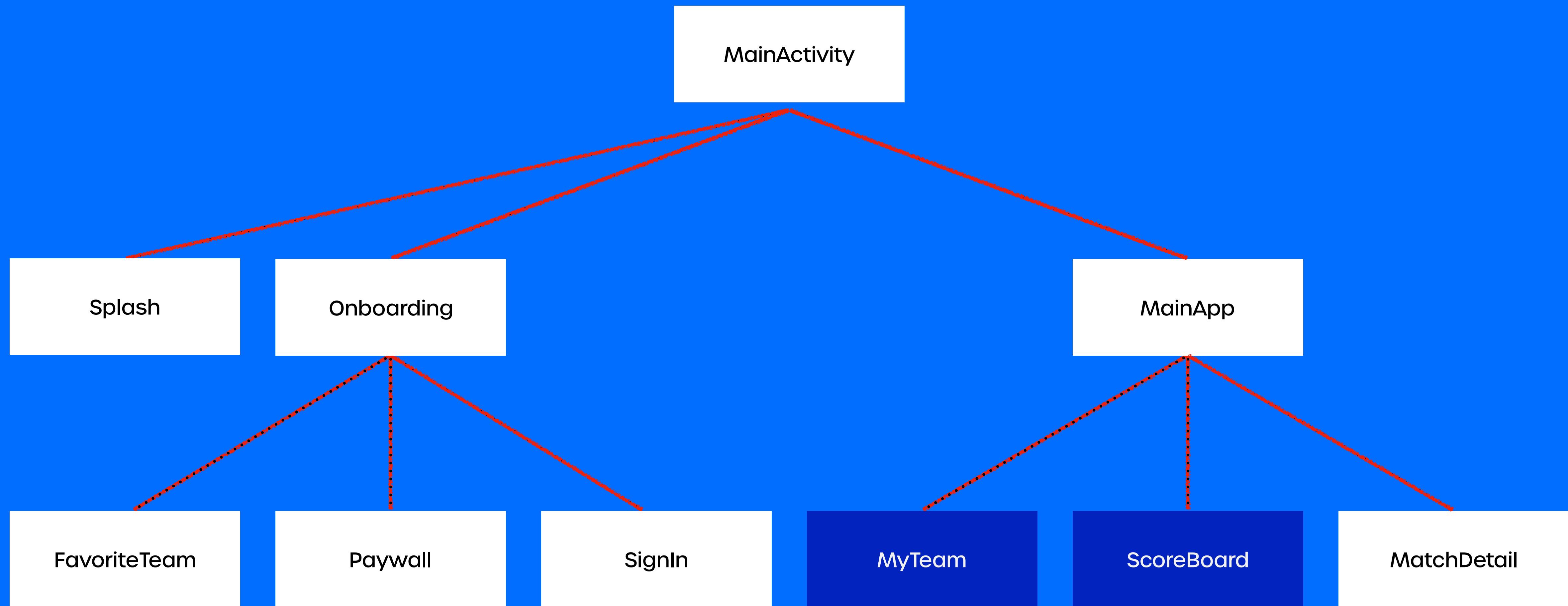
~~REDACTED~~

ScreenA

on Desktop View

on Desktop View

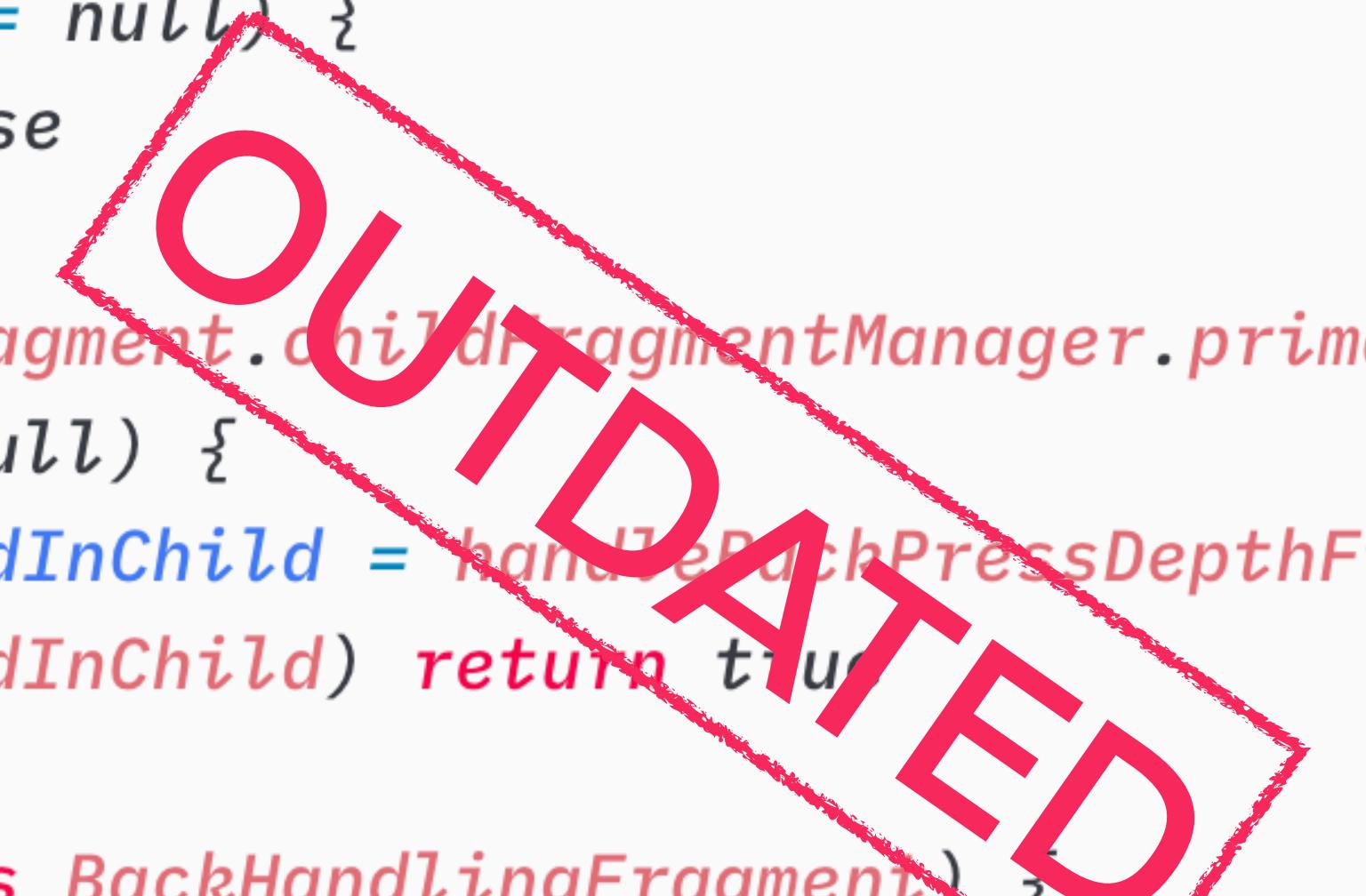
# Primary Navigation Fragment



# Required for nested back stacks

```
1 interface BackHandlingFragment {  
2     fun onBackPressed(): Boolean  
3 }
```

```
1 fun FragmentManager.handleBackPressedDepthFirst() =  
2     handleBackPressedDepthFirst(primaryNavigationFragment)  
3  
4 private fun handleBackPressedDepthFirst(fragment: Fragment?): Boolean {  
5     if (fragment == null) {  
6         return false  
7     }  
8     val child = fragment.childFragmentManager.primaryNavigationFragment  
9     if (child != null) {  
10        val handledInChild = handleBackPressedDepthFirst(child)  
11        if (handledInChild) return true  
12    }  
13    if (fragment is BackHandlingFragment) {  
14        return fragment.onBackPressed()  
15    }  
16    return false  
17 }
```



# OnBackPressedDispatcher

**As of activity:1.0.0-alpha06**

# Two Fragment navigation functions

```
1 fun FragmentManager.addToStack(  
2   container: Int,  
3   fragment: Fragment,  
4   backStackName: String? = null  
5 )
```

```
1 fun FragmentManager.replaceStack(  
2   container: Int,  
3   rootFragment: Fragment,  
4   vararg stack: StackEntry  
5 )  
6  
7 data class StackEntry(  
8   val fragment: Fragment,  
9   val tag: String? = null  
10 )
```

# Scalability

Single Activity by Example

Activity is the single entry point of each app

**Fragment is the single entry point of each feature**

Activity represents an entire user session

Fragments represent  
individual screens or flows

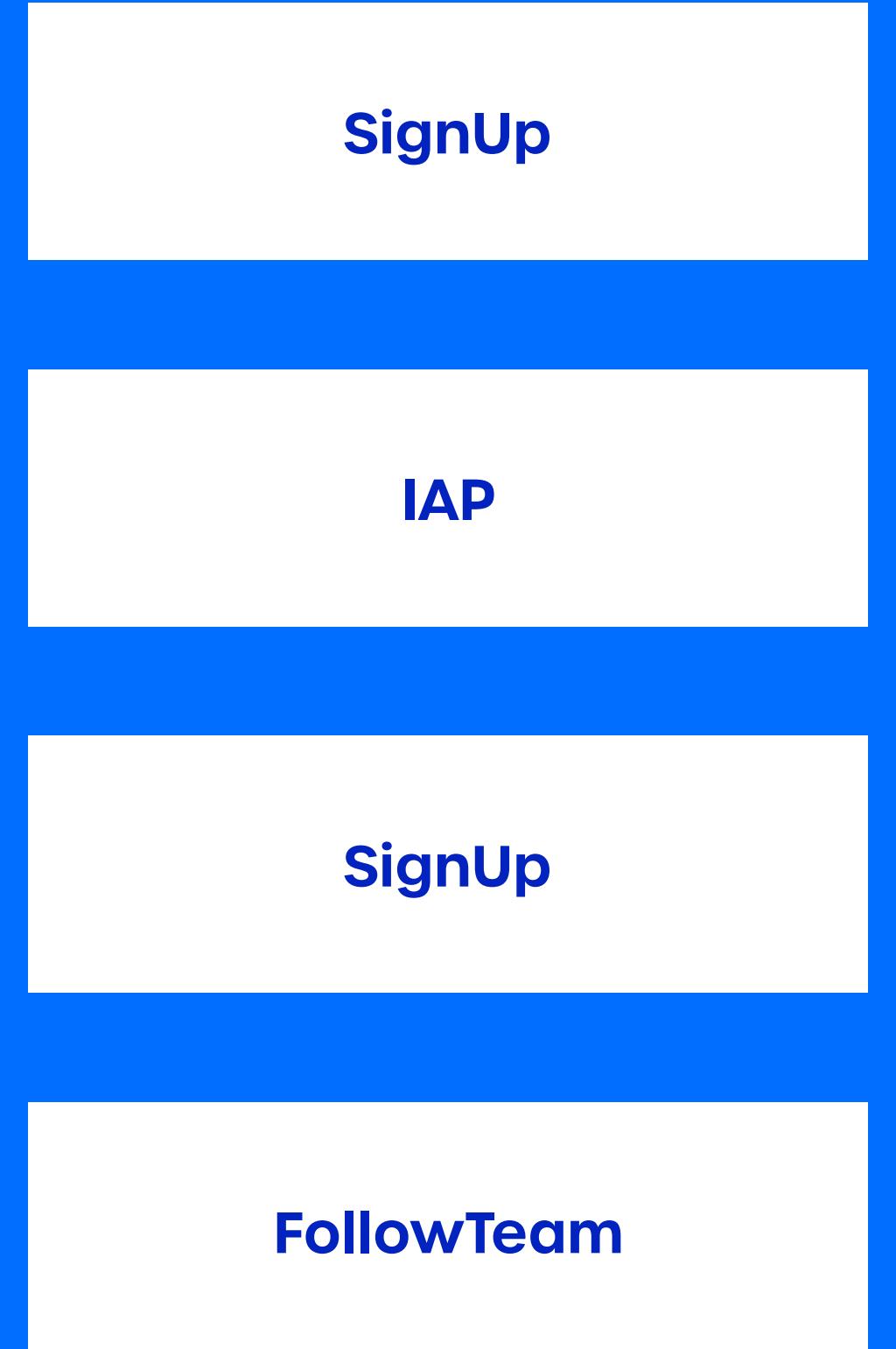
Fragments represent  
individual screens or flows

# Onboarding flow

Single Activity by Example

# IAP back behavior

Disney Streaming Services



9:00

Tuesday, Apr 2 | ☀ 8 °C



NHL Singl...

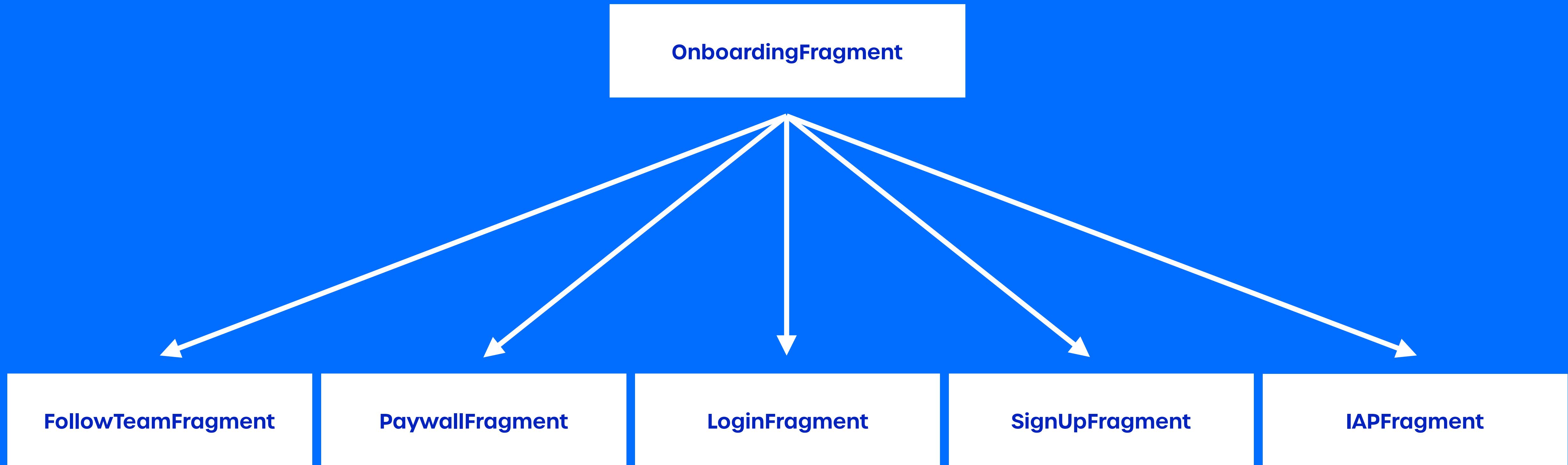


# Lots of different screens

# Complex back logic

# OnboardingActivity?

# Is onboarding done?



# replaceStack(OnboardingFragment())

```
class NHLActivity : Activity(), OnboardingHost
```

```
1 override fun onAttach(context: Context) {  
2     this.host = findParentThatImplements<OnboardingHost>()  
3     super.onAttach(context)  
4 }
```

```
1 interface OnboardingHost {  
2     fun onOnboardingComplete(account: NHLAccount)  
3 }
```

# Find interface in parents

```
1 fun <T> Fragment.findParentThatImplements(type: Class<T>): T {
2     return when {
3         parentFragment != null && type.isAssignableFrom(parentFragment.javaClass) ->
4             parentFragment as T
5         parentFragment != null ->
6             parentFragment.findParentThatImplements(type)
7         activity != null && type.isAssignableFrom(activity.javaClass) ->
8             activity as T
9         activity != null && type.isAssignableFrom(activity.application.javaClass) ->
10            activity.application as T
11        else -> throw IllegalStateException("None of the parents implement $type")
12    }
13 }
```

# What about that custom back behavior?

```
1 replaceStack(  
2     childFragmentManager.findFragmentByTag(TAG_FOLLOW_TEAM)  
3         ?: FollowTeamFragment(),  
4     TAG_FOLLOW_TEAM,  
5     StackEntry(SignUpFragment(), null)  
6 )
```

# Bottom Navigation

Single Activity by Example

(Almost) According to [material.io](#) (as of today)

Builds a stack  
within each tab

MatchDetail

Scoreboard

MyTeam

## Ottawa Senator

AWAY

7 St. Louis Blues 7

HOME

8 Nashville Predators 4

HOME

0 Boston Bruin 5

AWAY

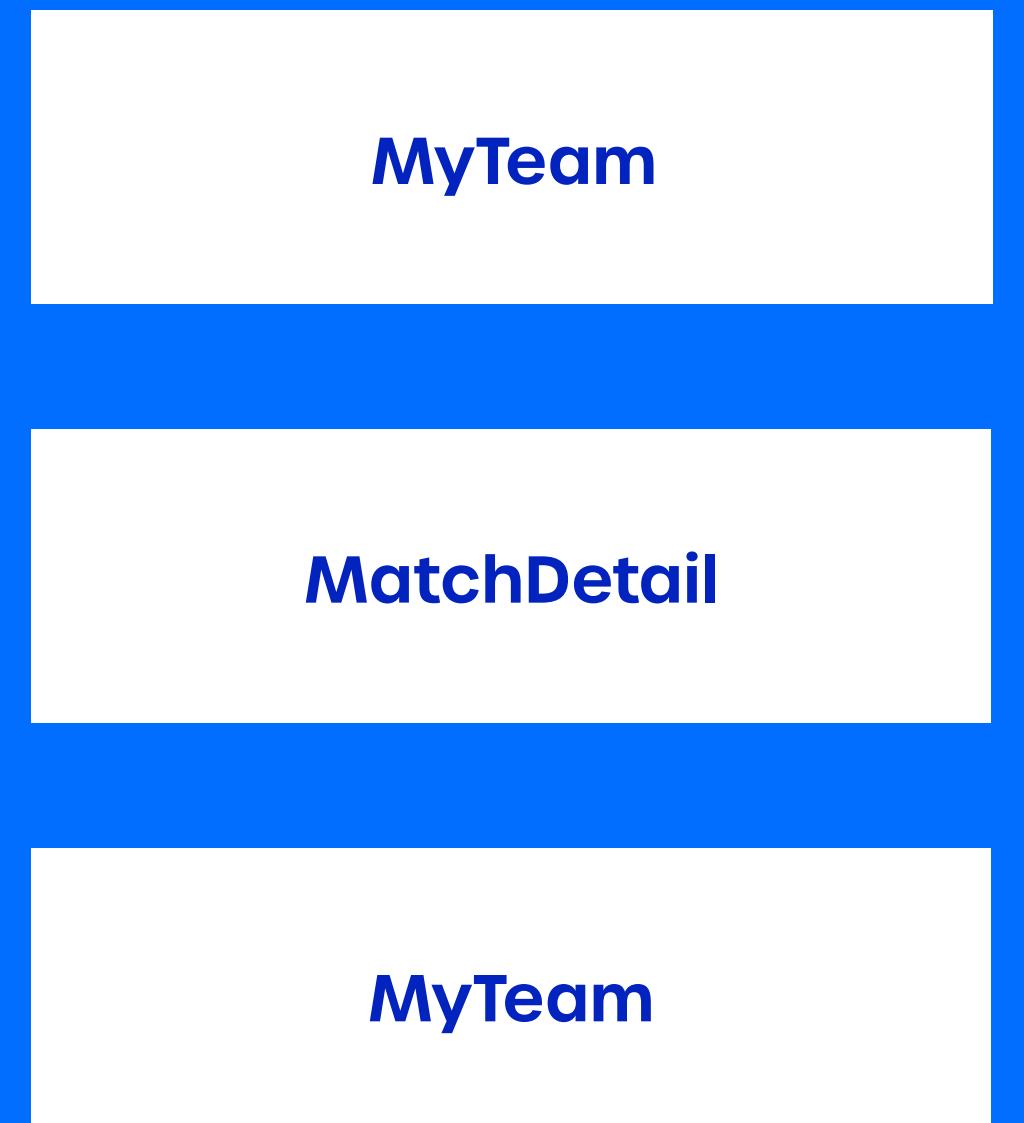
0 Pittsburgh Penguins 2

My Team

Scoreboard

Settings

# Switching tabs should reset the back stack



9:00      LTE      99%

## Ottawa Senator

HOME		AWAY
2	Winnipeg Jets	2
3	Vancouver Canucks	7
8	Montreal Canadiens	6
3	Tampa Bay Lightning	9

My Team      Scoreboard      Settings

Back on first tab closes the app

## Ottawa Senator

HOME

3 Philadelphia 8  
Flyers

AWAY

8 Tampa Bay 3  
Lightnin

HOME

1 Carolina 1  
Hurricanes

AWAY

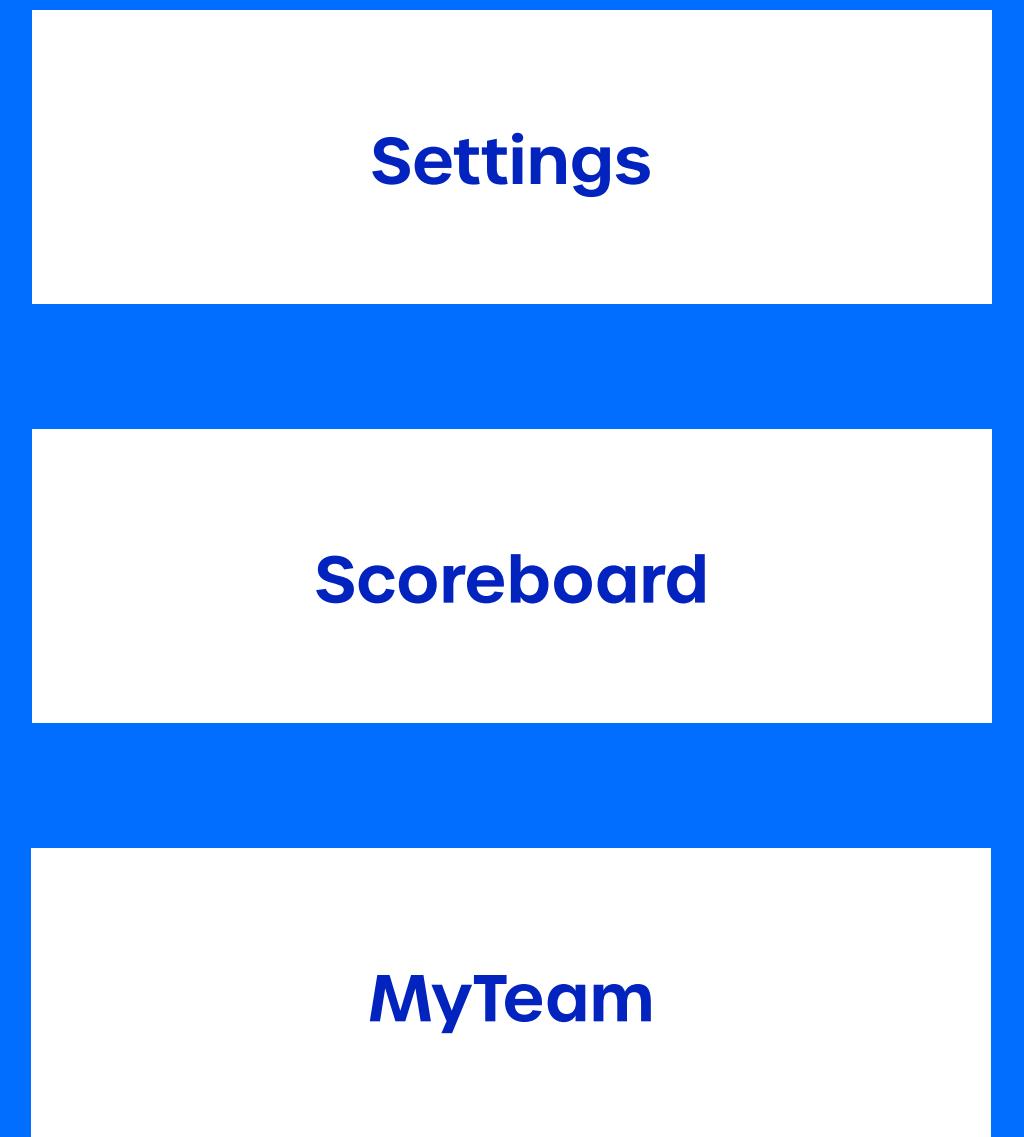
4 Colorado 4

My Team

Scoreboard

Settings

Pressing back  
on second and  
third  
top-level tab  
takes you to  
the first



# Settings

Push Notifications



Pressing tab item while on top level resets content

## Ottawa Senator

AWAY  
5 Anaheim Ducks 0

AWAY  
4 Carolina Hurricanes 6

HOME  
0 Florida Panther 8

HOME  
3 New York Islanders 2

My Team

Scoreboard

Settings

```
1 private fun startSecondaryTab(fragment: Fragment) {  
2     replaceStack(  
3         childFragmentManager.findFragmentByTag("MyTeam") ?: MyTeamFragment(),  
4         "MyTeam",  
5         fragment  
6     )  
7 }
```

# Startup checks

Single Activity by Example

# Needs to be done on every startup

# Pre fetch data

# Lapsed subscription check

# Fetch A/B Configs

# SplashActivity?

# SingleActivity.onCreate

```
1 class NHLActivity : DaggerAppCompatActivity() {  
2  
3     @Inject  
4     lateinit var accountRepository: AccountRepository  
5  
6     override fun onCreate(savedInstanceState: Bundle?) {  
7         super.onCreate(savedInstanceState)  
8         setContentView(R.layout.main_activity)  
9         if (savedInstanceState == null) {  
10             val cachedAccount = accountRepository.accountCache  
11             if (cachedAccount == null) {  
12                 startSplash()  
13             } else {  
14                 startMainAppFragment(cachedAccount)  
15             }  
16         }  
17     }  
18 }
```

```
1 class NHLActivity : DaggerAppCompatActivity() {  
2  
3     @Inject  
4     lateinit var accountRepository: AccountRepository  
5  
6     override fun onCreate(savedInstanceState: Bundle?) {  
7         super.onCreate(savedInstanceState)  
8         setContentView(R.layout.main_activity)  
9         if (savedInstanceState == null) {  
10             val cachedAccount = accountRepository.accountCache  
11             if (cachedAccount == null) {  
12                 startSplash()  
13             } else {  
14                 startMainAppFragment(cachedAccount)  
15             }  
16         }  
17     }  
18 }
```

# Key Takeaways & Tips

Single Activity by Example

# Nesting Fragments is good

# Every UI feature can be built in a Fragment

If a feature becomes too big,  
split your Fragment into multiple (child) Fragments

Fragments should never depend on their host

Fragments should depend on their host  
implementing an interface

# Always use Primary Navigation Fragment

# Always use setReorderingAllowed

Single Activity gives you  
full control over your back stack

# Fragment Caveats

Single Activity by Example

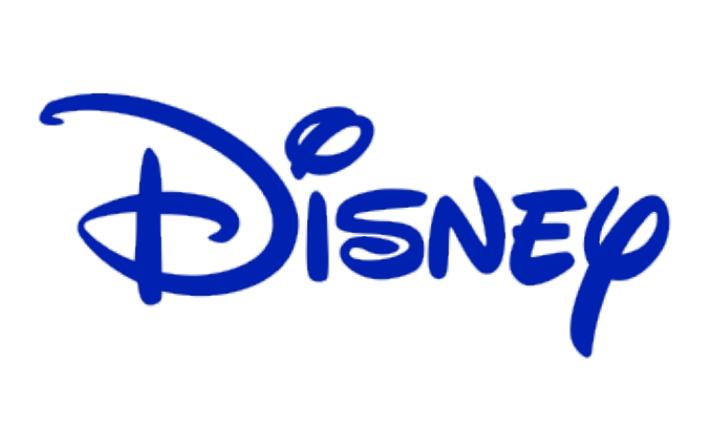
Only do initial navigation when  
savedInstanceState == null

**Don't mixup fragmentManager and  
childFragmentManager**

For navigation screens always:  
Add to back stack  
Or Replace the back stack

# Transitions support is...

[bit.ly/droidcon-it-single-activity](https://bit.ly/droidcon-it-single-activity)



streaming  
services

Thank you! Questions?