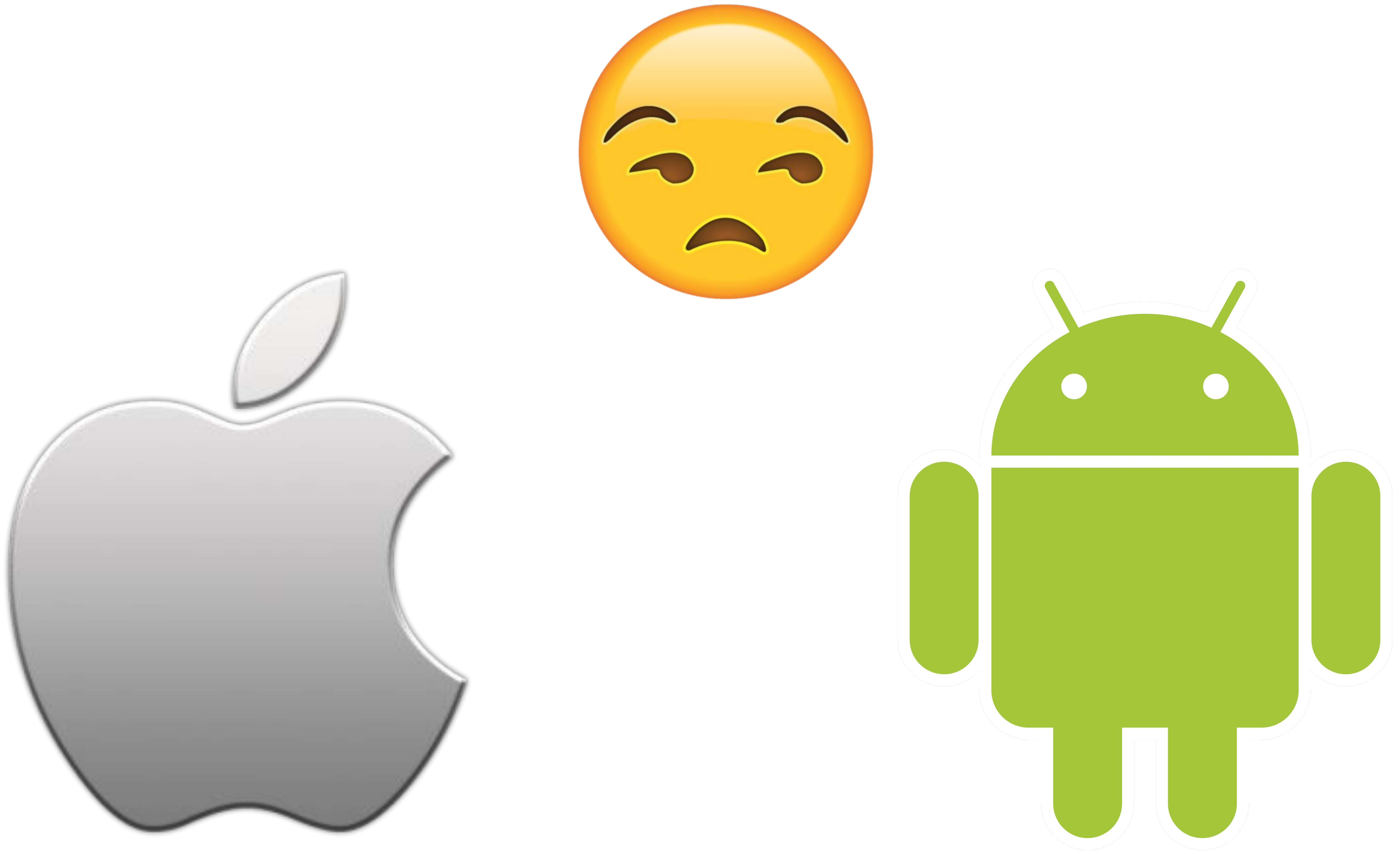
Partager du code entre le web et le mobile avec Kotlin Multiplatform

Martin Gagnon

But why?

CALL OF DUTY
WORLD AT WAR
ZOMBIES
PANDORA
BAND
STYLE.
COM /
yelp



BRACE YOURSELF
BECAUSE WEB, WATCH AND TV ARE COMING

titanium

PhoneGap

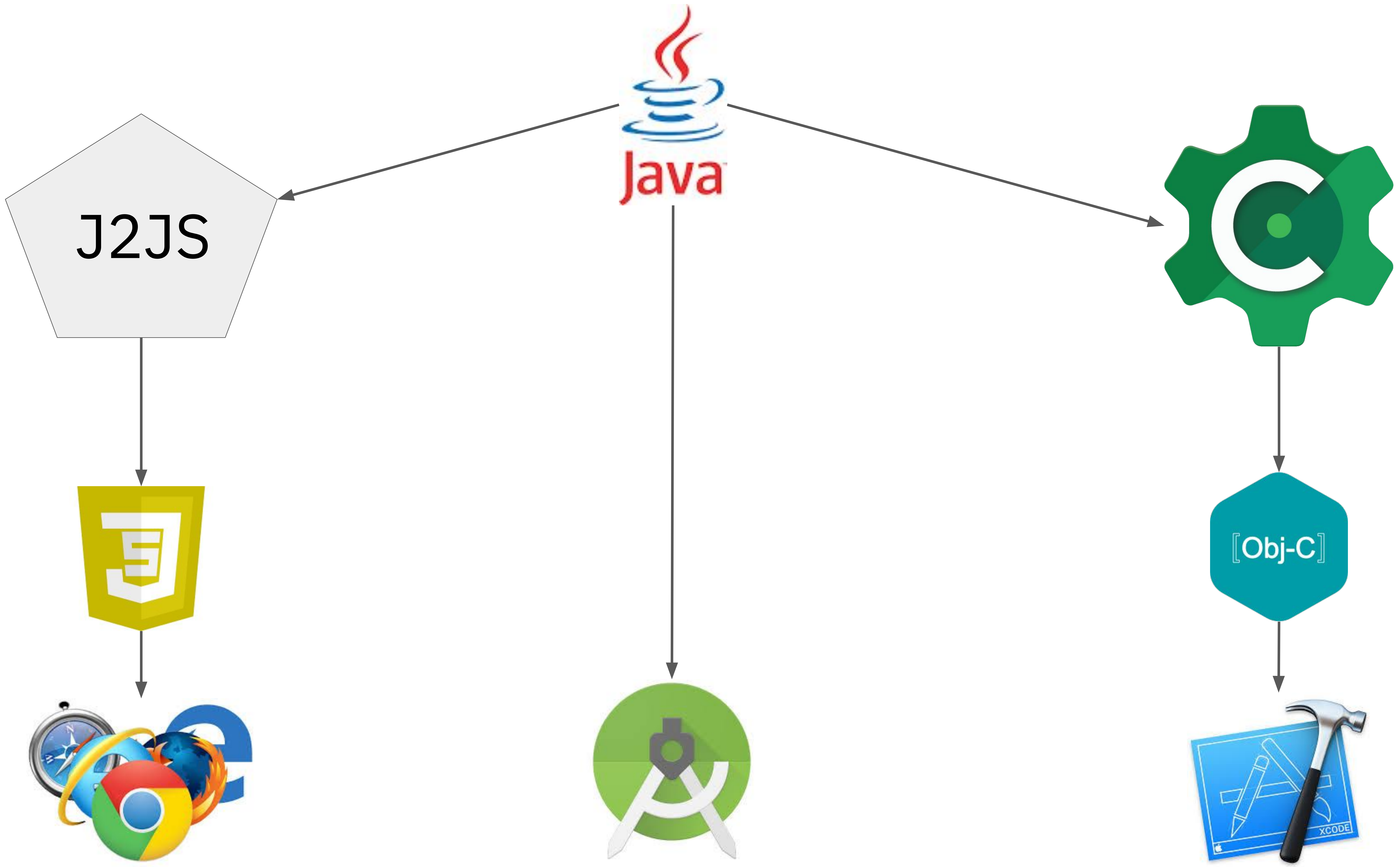
Flutter

APACHE CORDOVA™

No compromise on the product

We love platforms

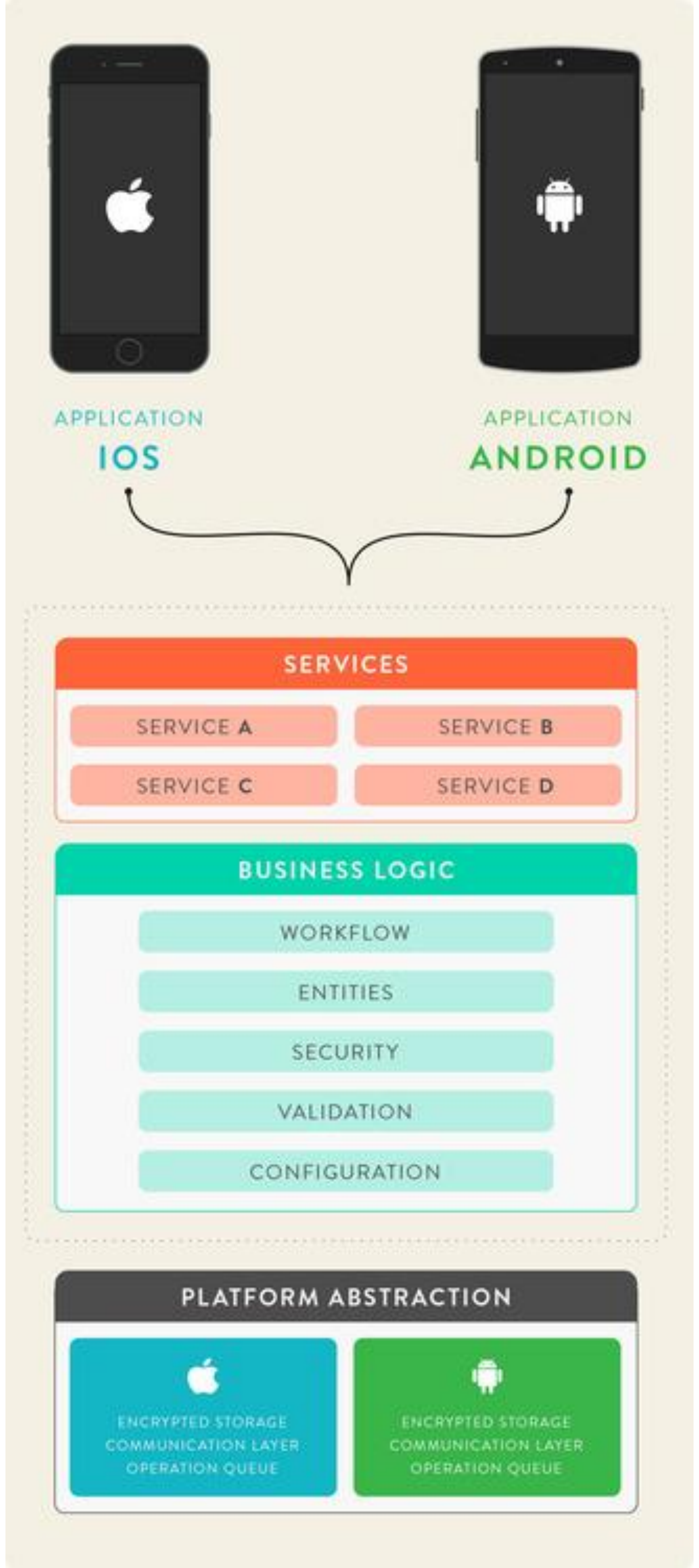
In house solution: SCRATCH

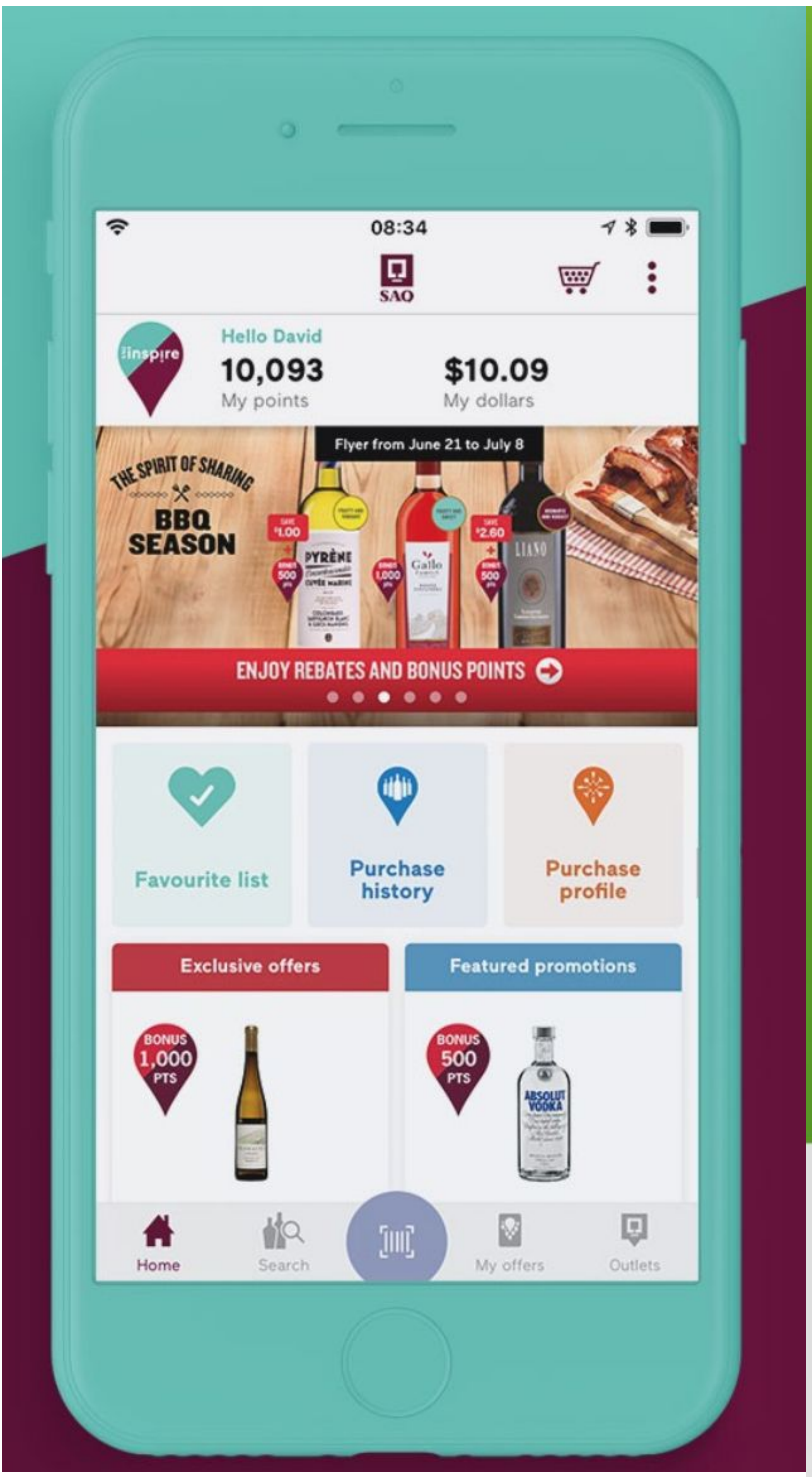

Java
J2JS
[Obj-C]
XCODE

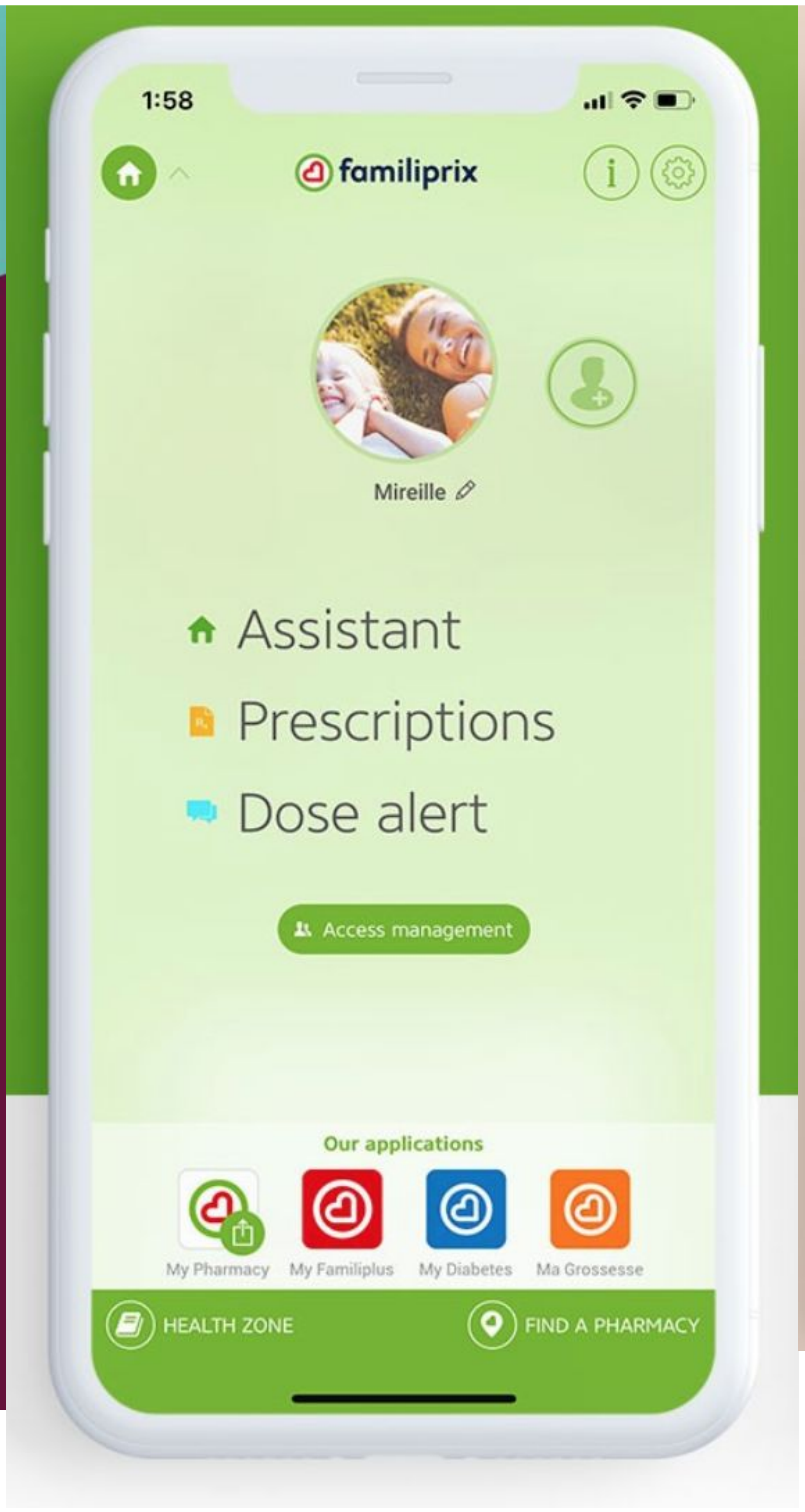
- UI/UX
- Animations

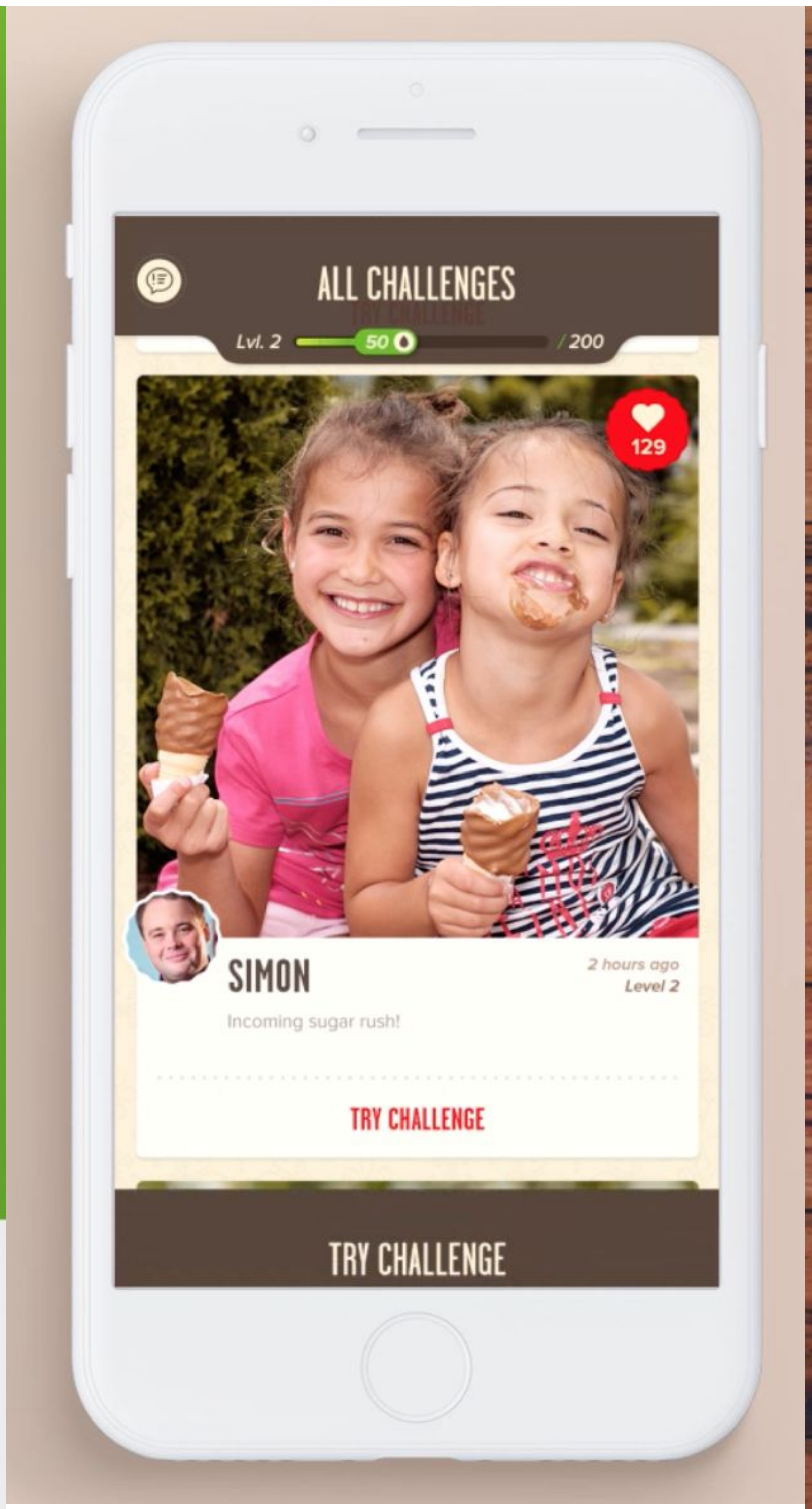
- Business Logic
- Models
- API Logic

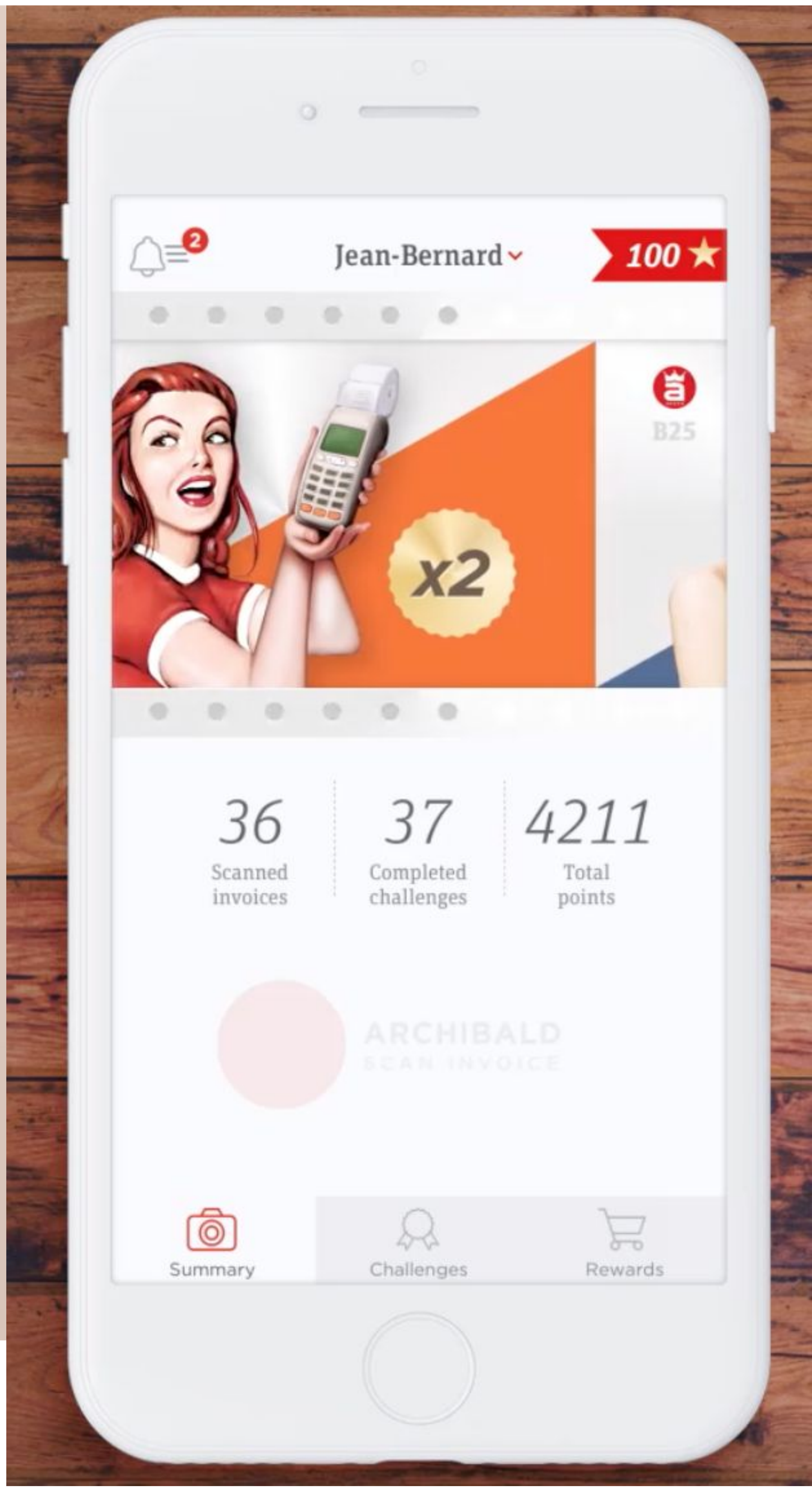
- HttpRequest
- Timers
- I/O




08:34
SAQ
inspire
Hello David
10,093
My points
$10.09
My dollars
Flyer from June 21 to July 8
THE SPIRIT OF SHARING
BBQ SEASON
ENJOY REBATES AND BONUS POINTS
Favourite list
Purchase history
Purchase profile
Exclusive offers
Featured promotions
BONUS 1,000 PTS
BONUS 500 PTS
Home
Search
My offers
Outlets


1:58
familiprix
Mireille
Assistant
Prescriptions
Dose alert
Access management
Our applications
My Pharmacy
My Familiplus
My Diabetes
Ma Grossesse
HEALTH ZONE
FIND A PHARMACY


ALL CHALLENGES
Lvl. 2
50
/200
129
SIMON
2 hours ago
Level 2
Incoming sugar rush!
TRY CHALLENGE
TRY CHALLENGE


2
Jean-Bernard
100
x2
B25
36
Scanned invoices
37
Completed challenges
4211
Total points
ARCHIBALD
SCAN INVOICE
Summary
Challenges
Rewards

Bell
Home
Guide
On Demand
Recordings
Search
Popular TV shows
Popular movies
Popular kids shows

Trending Live
Recent recordings
Continue enjoying
CTV News On Demand
Recently added
Series
MLB - Toronto Blue Jays at Tampa Bay Rays
Home
Guide
On Demand
Recordings
Settings
Help

KOTLIN MP
ME
SCRATCH

KMP outputs libraries

Shared

Kotlin.Native

JVM/Android

Kotlin.JS

Common module

```
package com.myapp.logger

expect class ConsoleLogger {
  fun log(string: String)
}
```

Common module

Kotlin.JS module

```
actual class ConsoleLogger {
  actual fun log(valueToLog: String) =
     console.log(valueToLog)
}
```

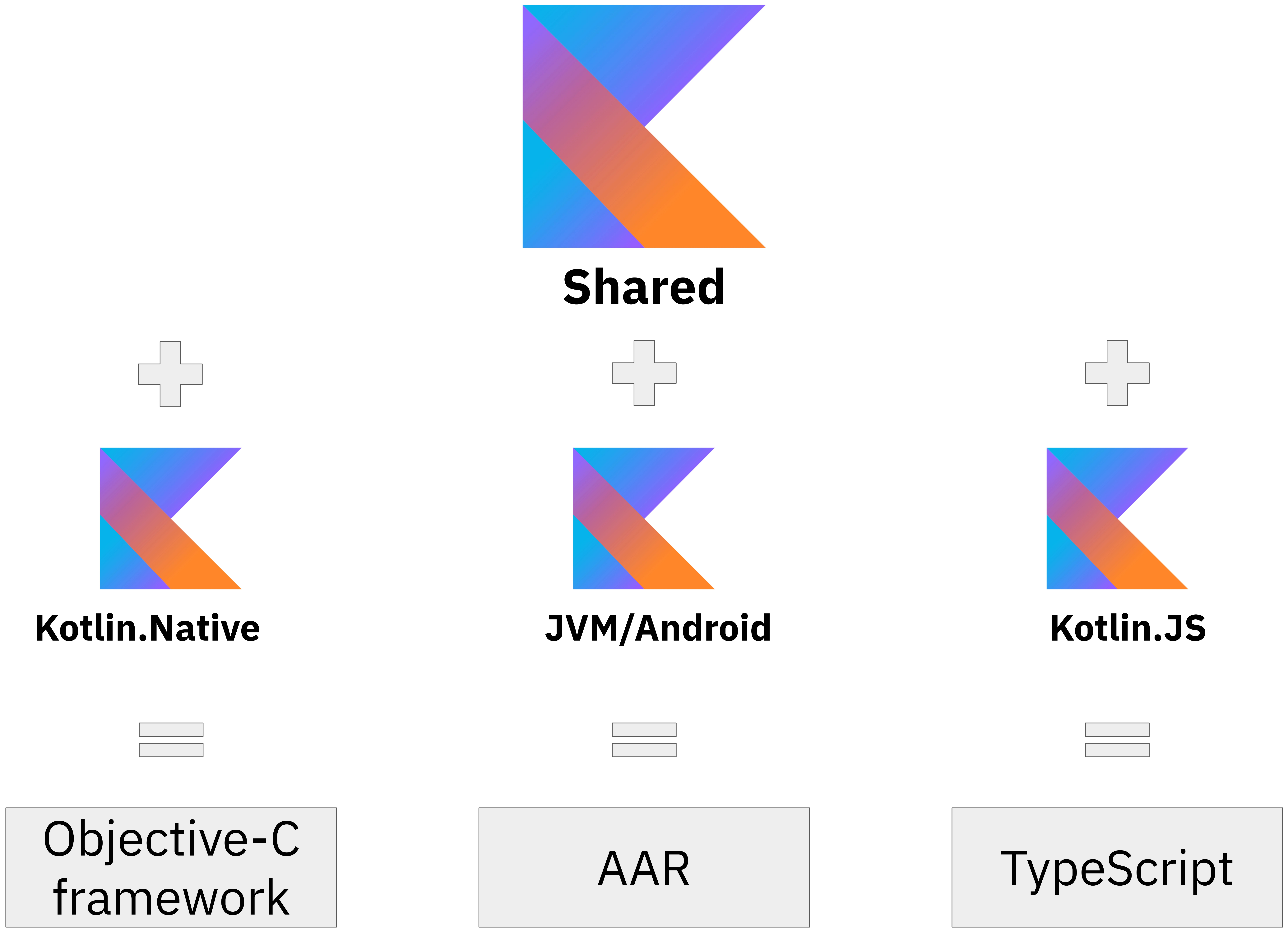
Android module

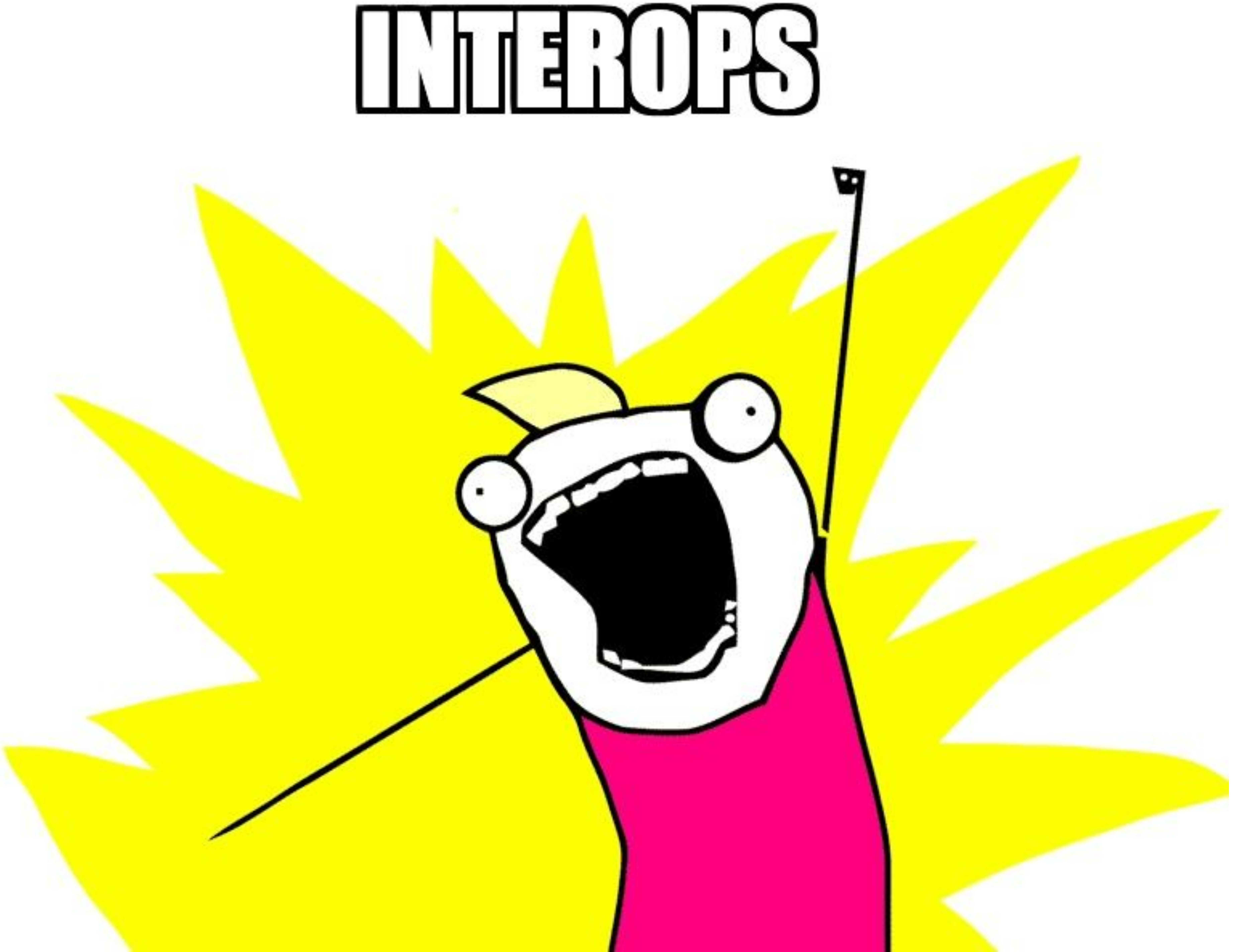
```
actual class ConsoleLogger {
  actual fun log(valueToLog: String) =
    Logger.getGlobal()
      .log(Level.INFO, valueToLog)
}
```

Kotlin.Native module

```
actual class ConsoleLogger {
  actual fun log(valueToLog: String) =
    NSLog(“$valueToLog”)
}
```

```
class SharedCode {
  init {
    val consoleLogger = ConsoleLogger()
    consoleLogger.log(“sharedCode initialized”)
  }
}
```


Shared
Kotlin.Native
JVM/Android
Kotlin.JS
Objective-C framework
AAR
TypeScript


INTEROPS

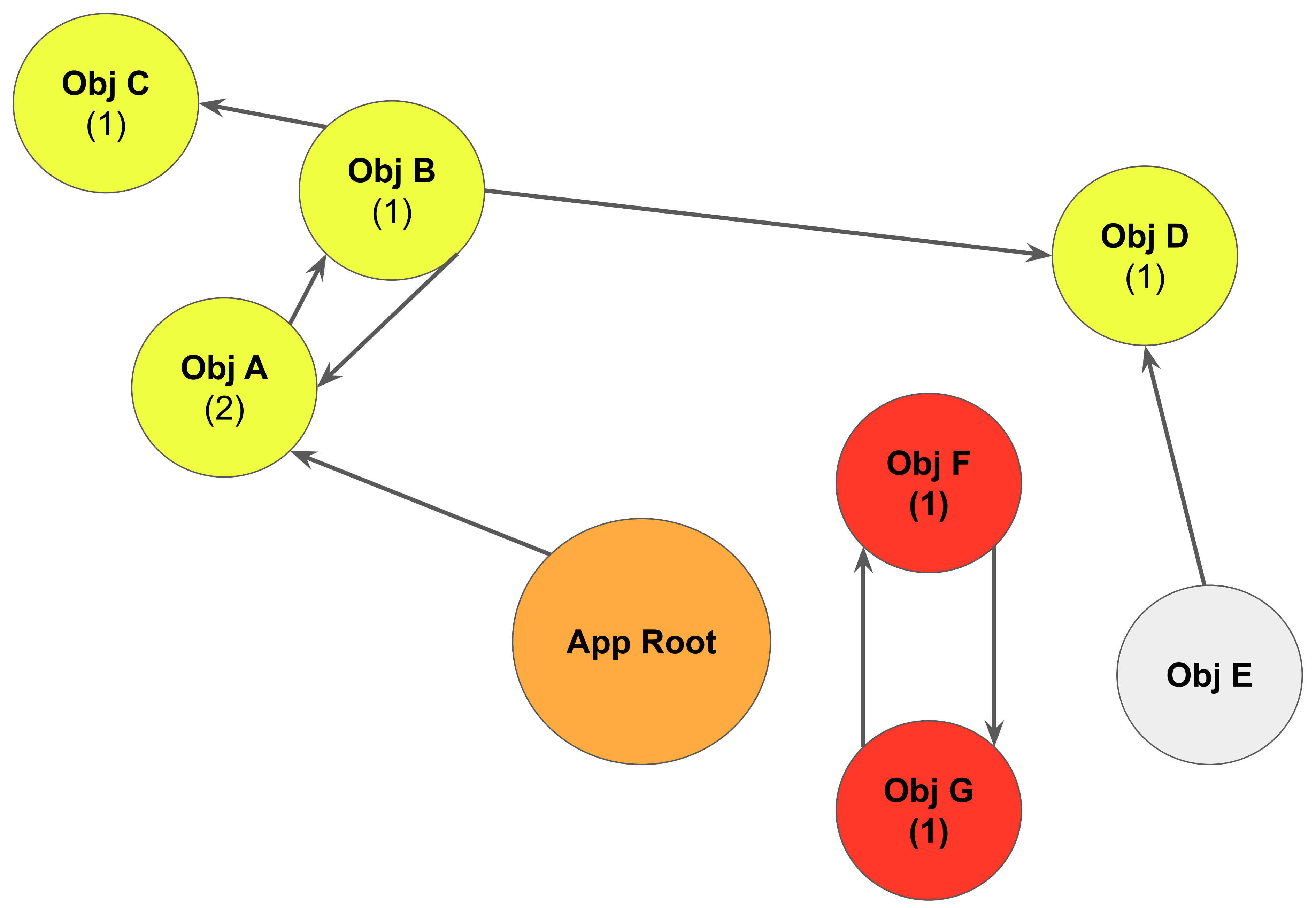
```
import SharedFramework

func printFooBar() {
  val fooBar = FooBar().getFooBar()

  print(fooBar.foo)
}
```

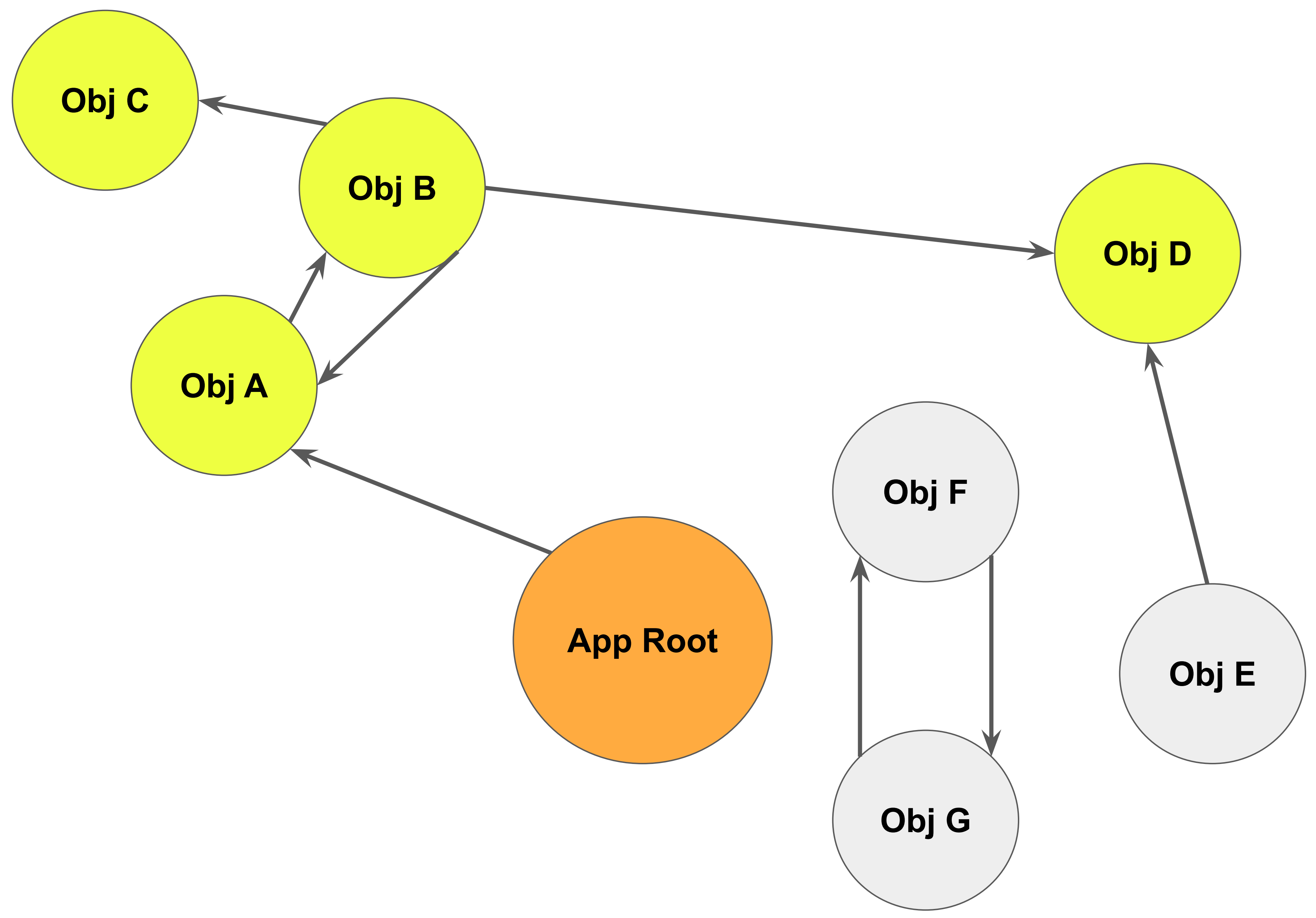
```
import {FooBar} from 'path/to/shared-library';


const {foo, bar} = FooBar.getFooBar();

console.log(foo)
console.log(bar)
```

Memory management

Reference Counting

Garbage collection

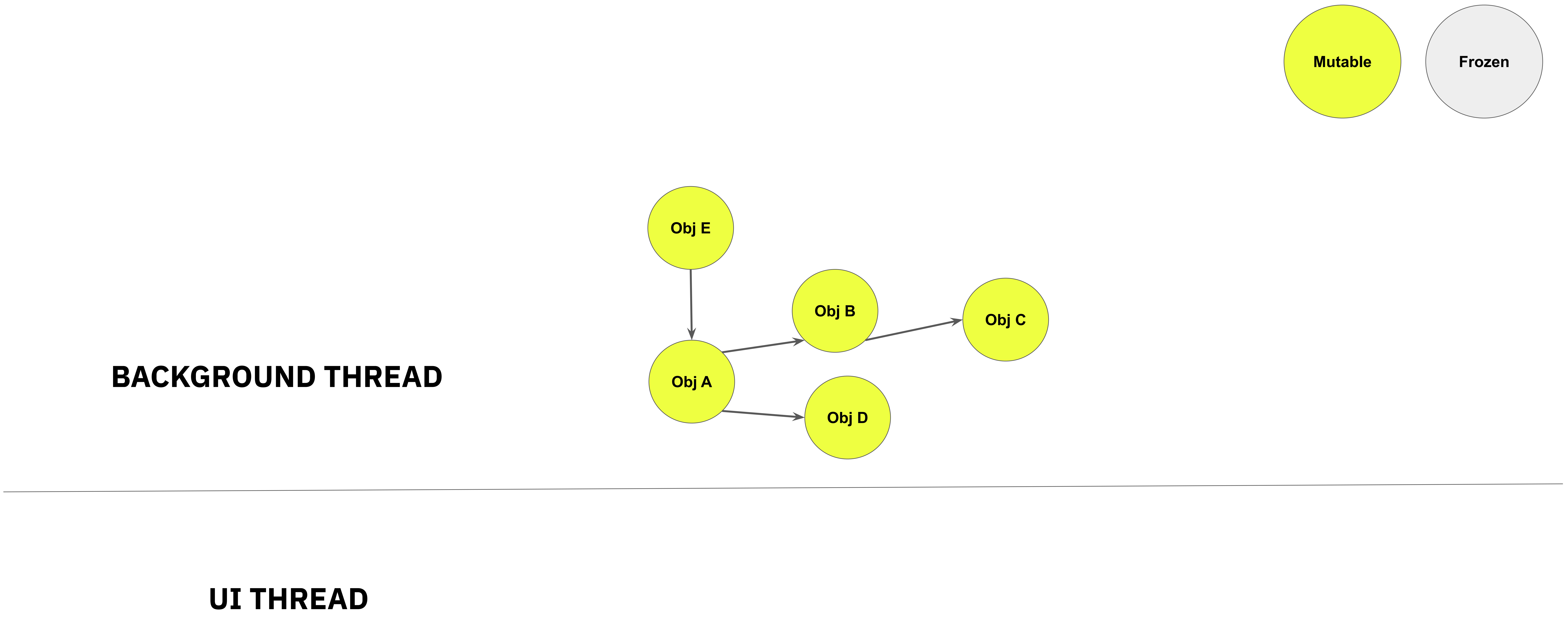


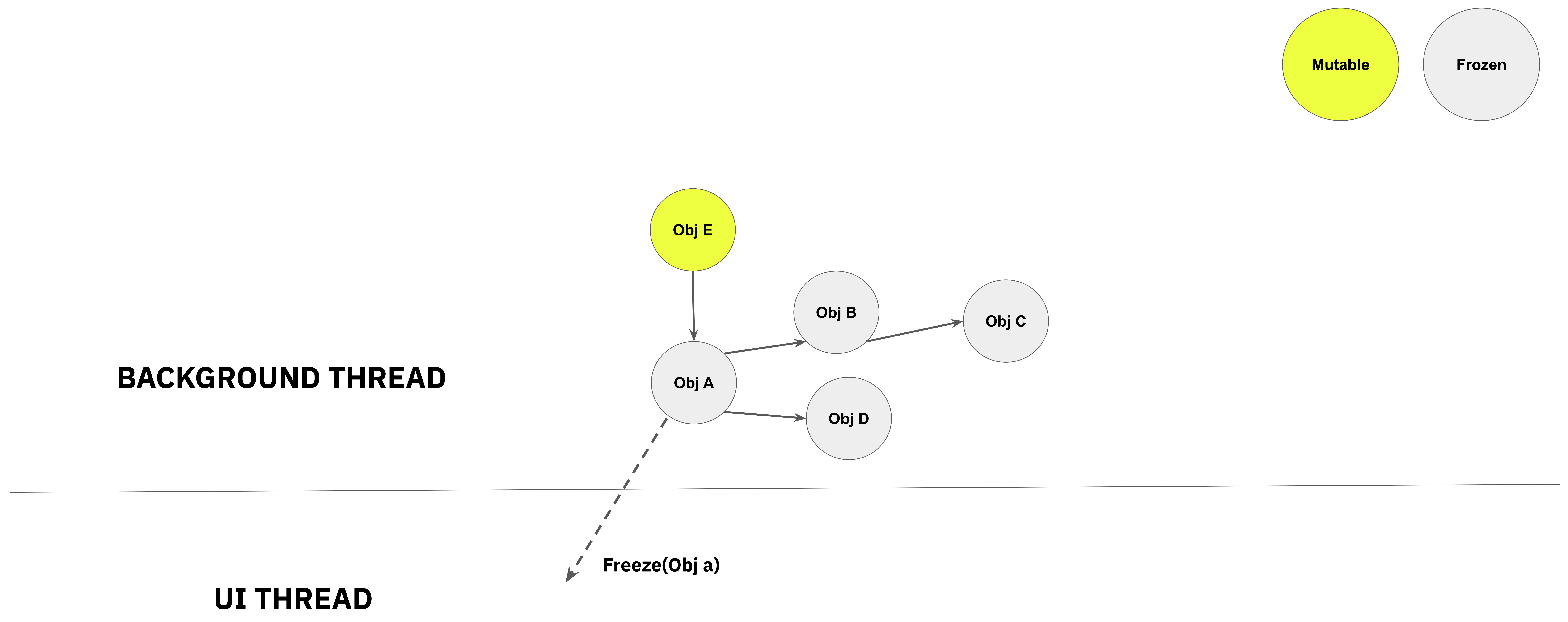


WHEN YOU HAVE GARBAGE COLLECTION
AND YOUR AN IOS DEVELOPER
imgflip.com

Multithreading before 1.6.10

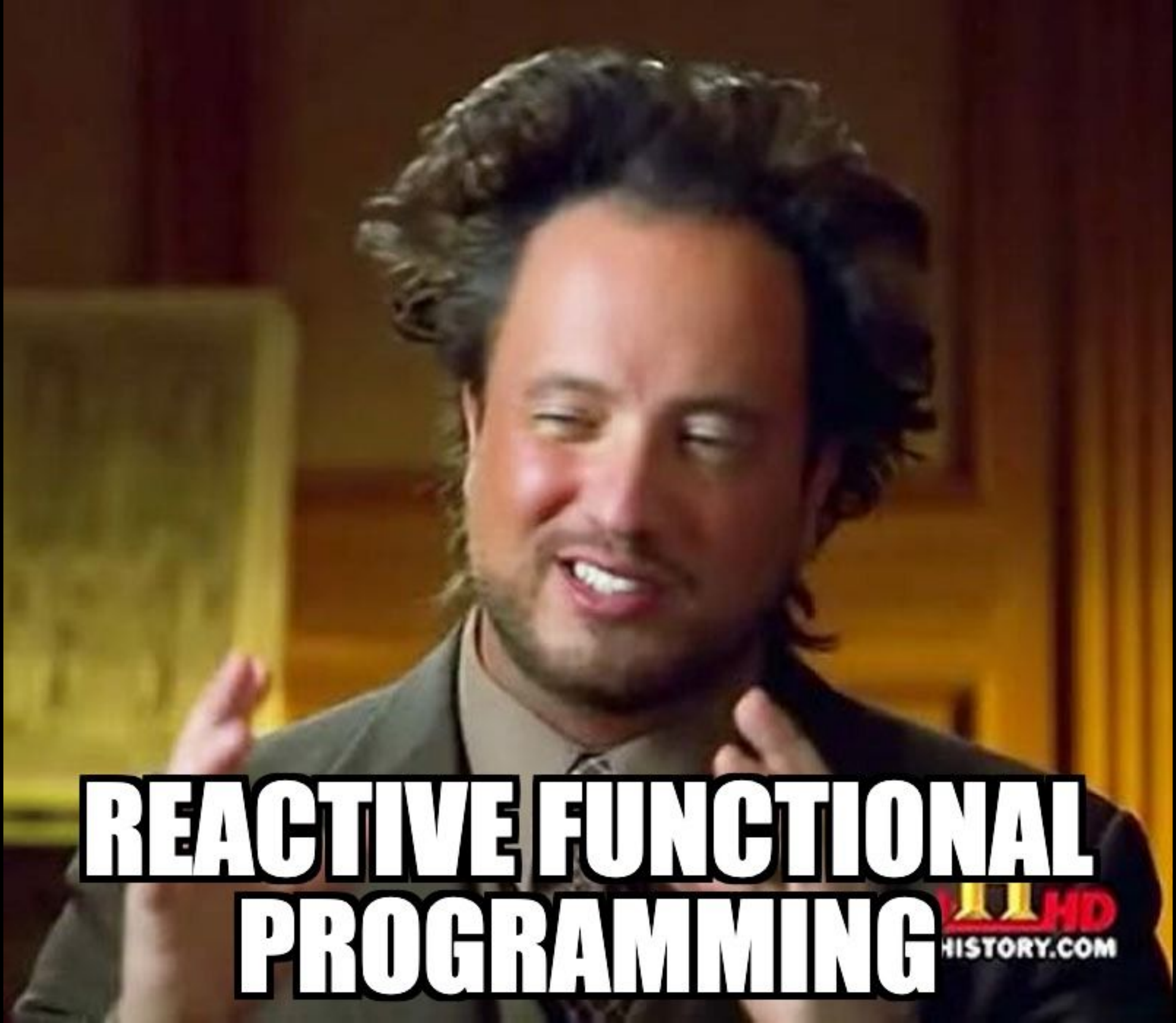
Kotlin/Native implements **strict mutability checks**, ensuring the important invariant that the object is either **immutable** or accessible from the **single thread** at that moment in time (`mutable XOR global`).


Mutable
Frozen
Obj E
Obj B
Obj C
BACKGROUND THREAD
Obj A
Obj D
UI THREAD


Mutable
Frozen
Obj E
Obj B
Obj C
BACKGROUND THREAD
Obj A
Obj D
Freeze(Obj a)
UI THREAD

Class AtomicReference can be used to publish the changed frozen state to other threads, and so build patterns like shared caches.

```
class Foo {
  val mutableString = AtomicReference<String>()

  …

  fun mutateString(newValue: String) {
    val oldValue = mutableString.value
    mutableString.compareAndSet(oldValue, freeze(newValue))
  }
}
```


REACTIVE FUNCTIONAL PROGRAMMING
HD
HISTORY.COM

- Foundation (Date, Timers, Queues)
- Streams (ReactiveX)
- Http
- Graphql
- KWord (Translation)
- ViewModels
- DeclarativeViewModels
- DataSources (Storage, Cache)
- Bluetooth

Trikot

https://github.com/mirego/trikot

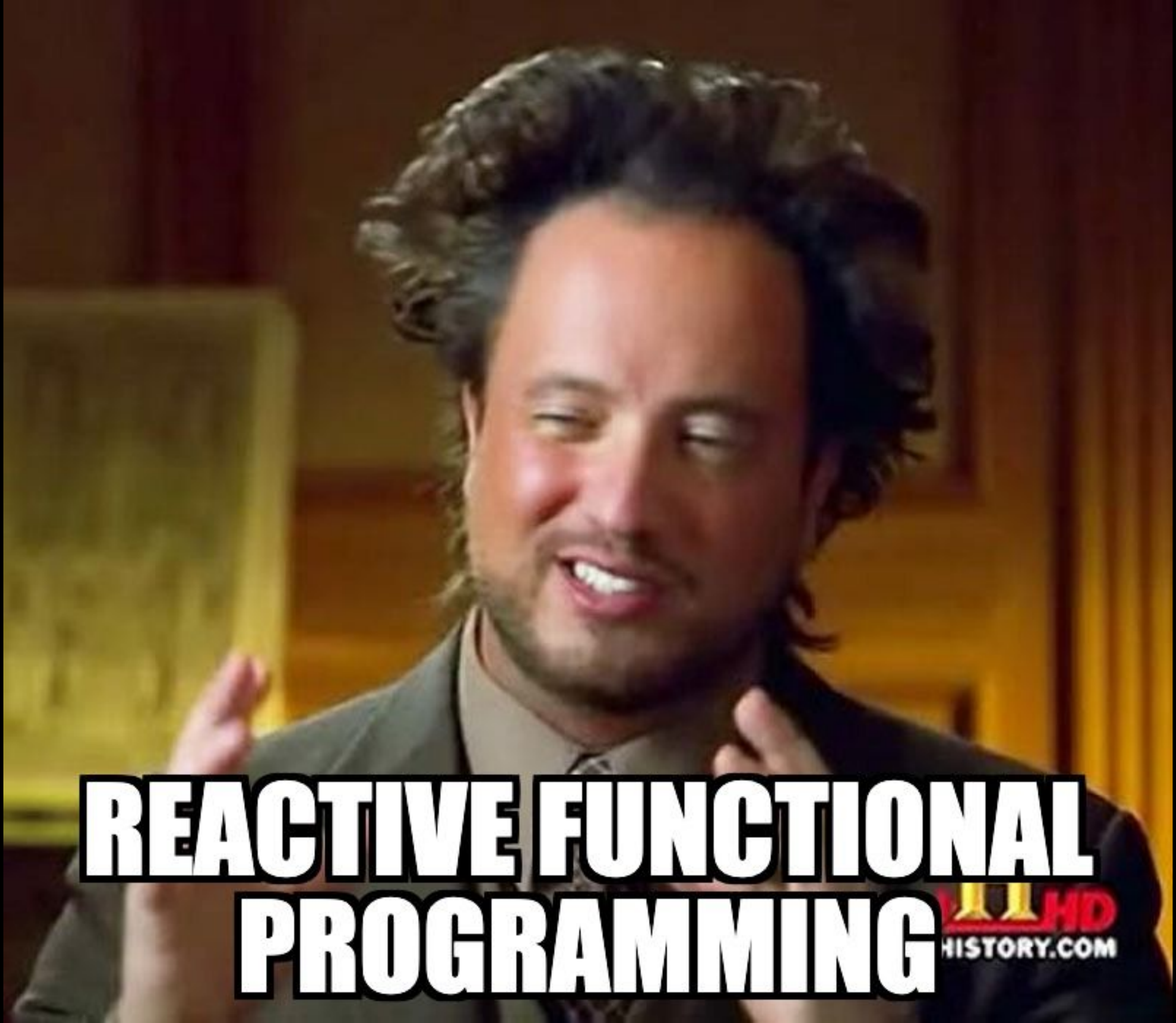
Multithreading since 1.6.10

The new Kotlin/Native automatic memory manager lifts the existing restrictions on object sharing between threads and provides **fully leak-free concurrent programming primitives** that are safe and **don’t require any special management or annotations** from the developers.

MY HYPE LEVEL
IS OFF THE CHARTS!!!!!
memegenerator.net

KOTLIN COROUTINES

```
suspend fun getFeed(): String {
    return httpClient.get(FEED_URL).bodyAsText()
}
```


REACTIVE FUNCTIONAL PROGRAMMING
HD
HISTORY.COM

```kotlin
val feedString = hasInternet
  .flatMapLatest { hasInternet ->
      when {
        !hasInternet -> flowOf("")
        hasInternet -> flow { emit(getFeed()) }
      }
}
```

```kotlin
hasInternet: Flow<Bool>
```

```kotlin
suspend fun getFeed(): String
```

```
feedString.watch { [weak self] self?.myLabel.text = $0 }
```

```
val stringState = feedString.collectAsState()
```

```
 useEffect(() => {
  obs = feedString.observeState(setFeedString)
  return () => obs.cancel()
}, …)
```

Since 2019

10 + apps launched

Some of them were existing apps

35 + devs

Around 40 mobile devs developed with KMP

200k + users

More than 200k users have downloaded and used our apps

8 + platforms

iPhone, Android, iPad, Apple TV, Android TV, Fire Tv, Ember, React

ONE DOES NOT SIMPLY
PRESS COMPILE

Where To Begin

https://github.com/Kotlin/kmm-production-sample

Questions?

→ mgagnon@mirego.com

→ Martin Gagnon - https://kotlinlang.slack.com/

→ vie.mirego.com