

Milan, 14 June 2018



agile venture
milan kotlin community conf



Milan
Kotlin
Community
Conf

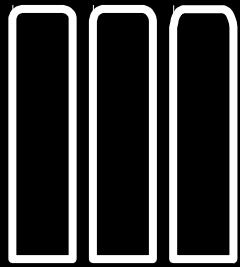
Multiplatform Kotlin Projects

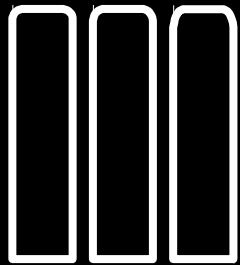
Victor Kropp
@kropp

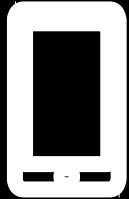
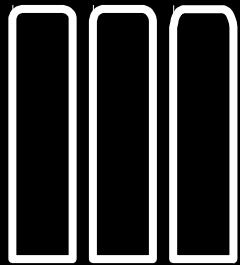


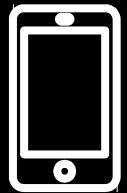
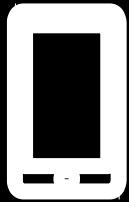
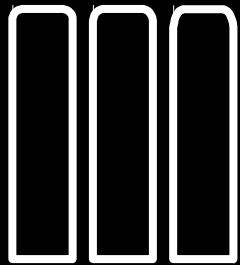
@KotlinCommConf

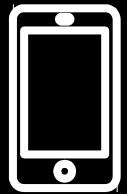
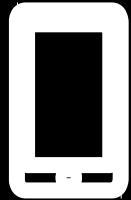
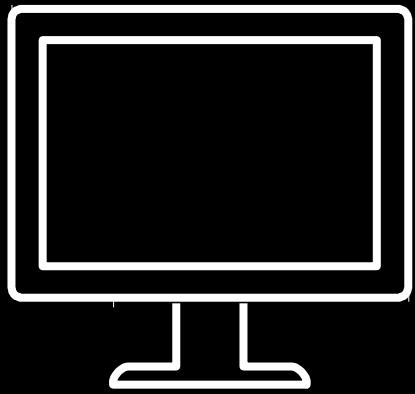
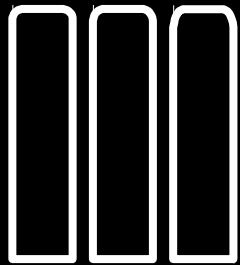
#KCC18











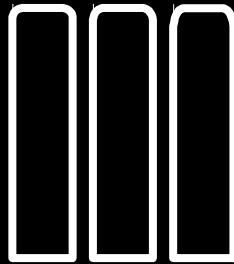
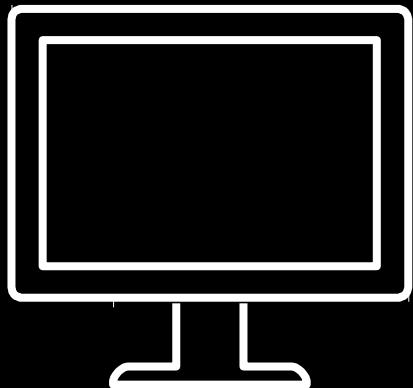
C

C++

C#

Objective-C

Swift



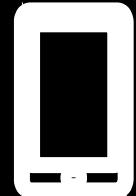
Java

Python
Ruby

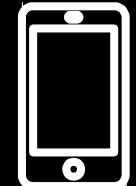
JavaScript
TypeScript



Java
Kotlin



Objective-C
Swift



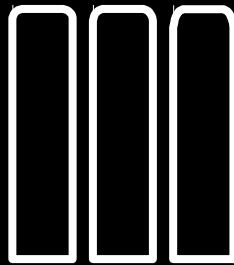
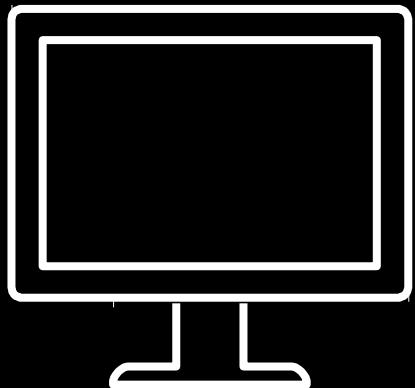
C

C++

C#

Objective-C

Swift



Java

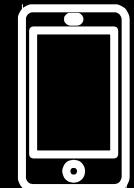
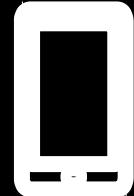
Python
Ruby

JavaScript
TypeScript



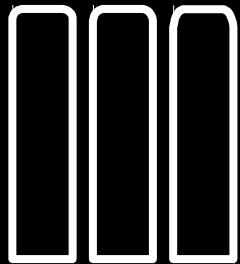
Java

Objective-C
Swift

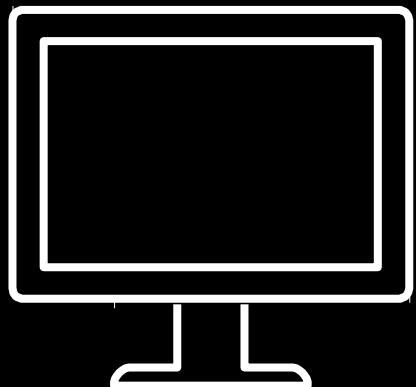




Kotlin



Kotlin

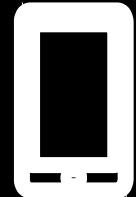


Kotlin

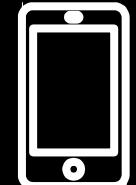


Kotlin

Kotlin



Kotlin



Using the same language across the platforms

Everyone on the team speaks the same language

Single team working on all apps

Simplify full-stack development

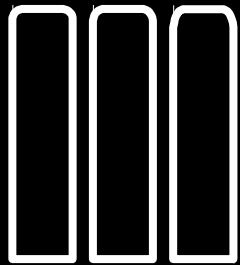
**Share code
between platforms**

Write once,
run everywhere

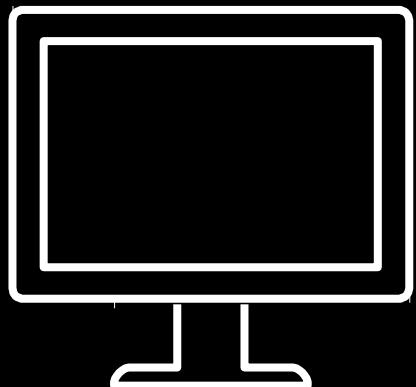


Write once,
debug everywhere





Kotlin

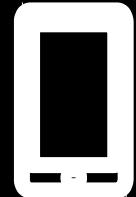


Kotlin

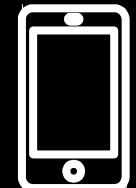


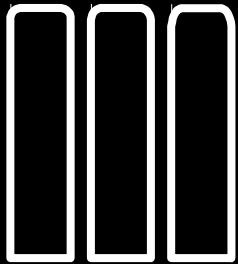
Kotlin

Kotlin

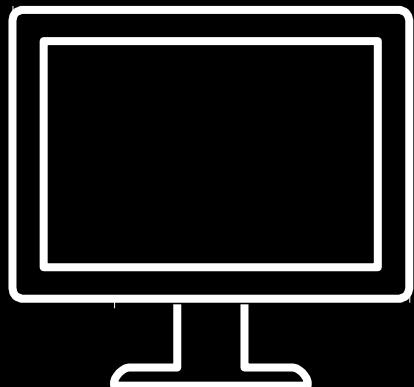


Kotlin



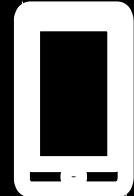


Kotlin/Native



Kotlin/JVM

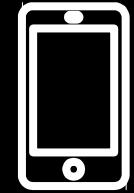
Kotlin/JVM



Kotlin/JS



Kotlin/Native



Code sharing

Share data structures

Share business logic

Code sharing

Share data structures

Share business logic

Share tests!

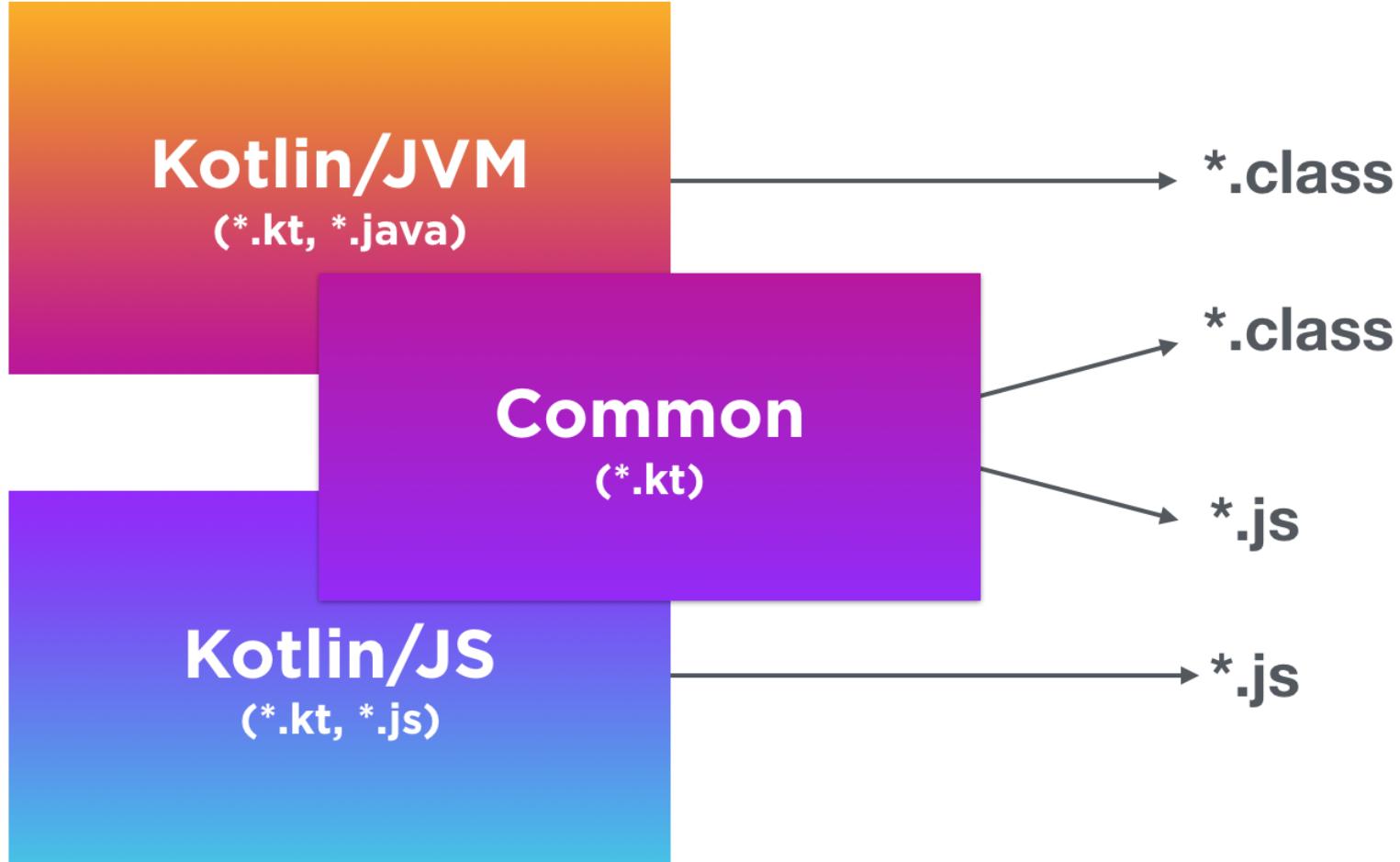
Code sharing

Share data structures

Share business logic

Share tests!

Do NOT share UI



Common module

```
apply plugin: 'kotlin-platform-common'

dependencies {
    compile "org.jetbrains.kotlin:kotlin-stdlib-common:$version"
}
```



Platform module

```
apply plugin: 'kotlin-platform-jvm'

dependencies {
    compile "org.jetbrains.kotlin:kotlin-stdlib:$version"
    expectedBy project(":mp-common")
}
```



expect/actual

```
expect class Foo {  
    fun baz()  
}
```

```
actual class Foo {  
    actual fun baz() {}  
}
```



expect/actual

```
expect class Foo(bar: String) {  
    fun baz()  
}
```

```
actual class Foo actual constructor(val bar: String) {  
    actual fun baz() {}  
}
```



typealias

expect annotation class Test

actual typealias Test = org.junit.Test



typealias

```
expect class BigDecimal {  
    fun divideAndRemainder(d: BigDecimal): Array<BigDecimal>  
}
```

```
actual typealias BigDecimal = java.math.BigDecimal
```



Sample project

- ▶  mp-common
- ▶  mp-js
- ▶  mp-jvm
- ▶  server
- ▶  web

Show me the code!



Code



<https://github.com/kropp/kotlin-multiplatform-sample>

Common modules

- ▶ **Coroutines**
- ▶ **kotlin.test**
- ▶ **kotlinx.serialization**
- ▶ **kotlinx.html**

And many more libraries in the future

Recap

- ▶ **Kotlin allows you to write the whole application in a single language**
- ▶ **Reuse business logic and data structures**
- ▶ **Integrate with respective platform**
- ▶ **Multiplatform projects for JVM & JS available now
Kotlin/Native coming soon**

Thank you!

Victor Kropp
@kropp
victor.kropp.name

Questions?

Victor Kropp
@kropp
victor.kropp.name