

Kotlin Use Cases



Markus Neuhoff
Engineering Leader and Developer

**Fully interchangeable
with Java code**

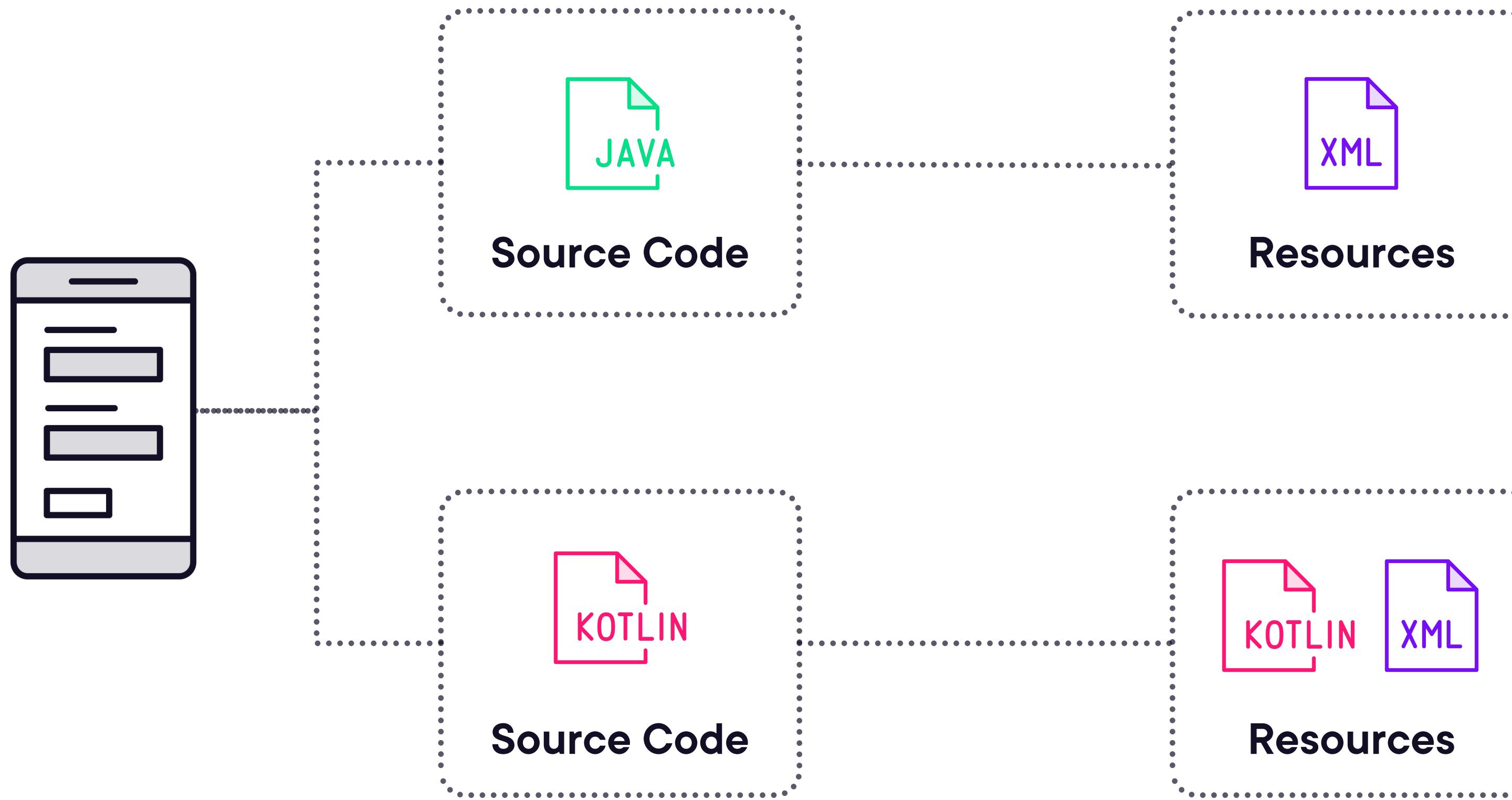
**Google's preferred
approach since 2019**

**Growing number of Kotlin
only libraries**

Kotlin for Android



Android Development



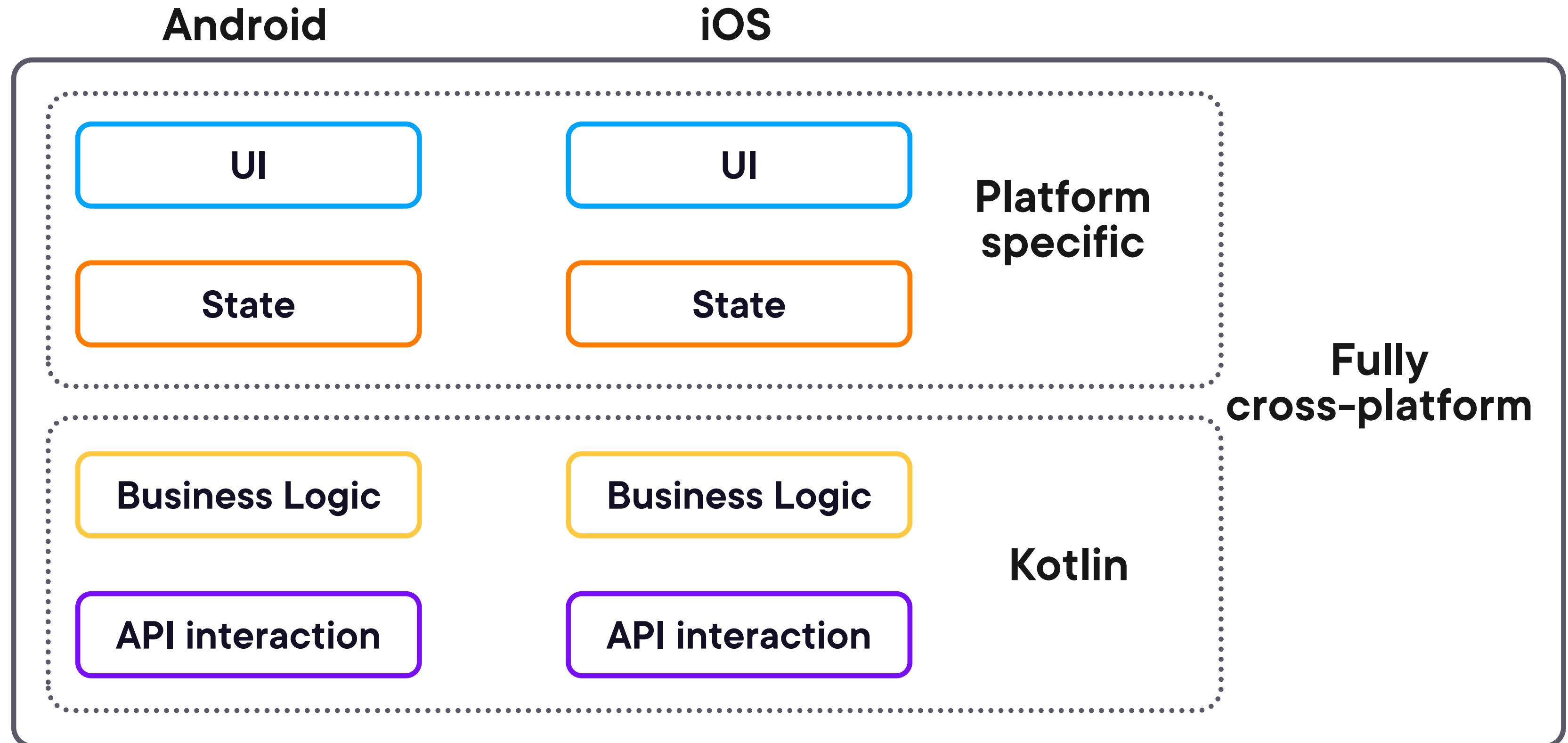
Developing Android Applications with Kotlin: The Big Picture



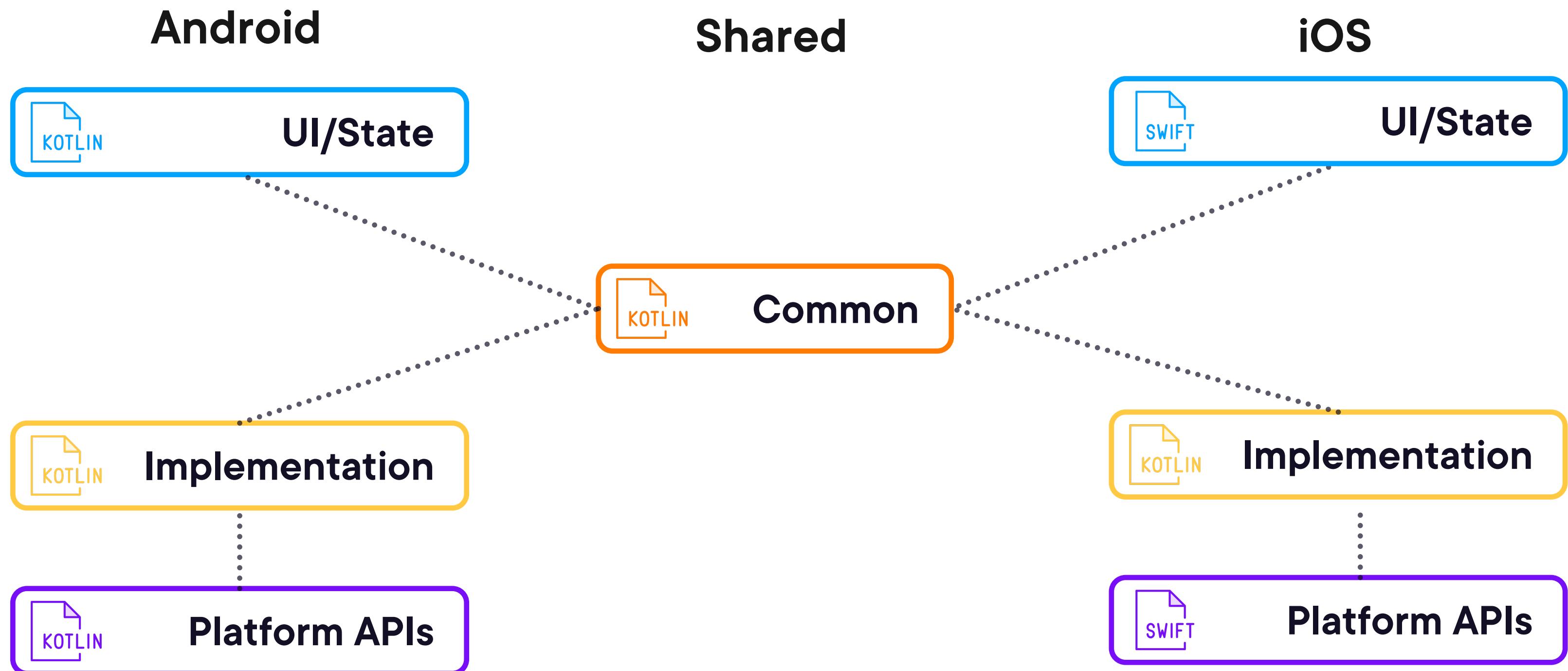
Cross-platform Mobile Development



Mobile Development



Implementation Path



Cross-platform Mobile Use Cases

Network connectivity

Analytics

Company specific
business logic

Common utilities

Database
interactions

Encryption





Kotlin for JavaScript



Kotlin for JavaScript Implementations

Cross-platform Library

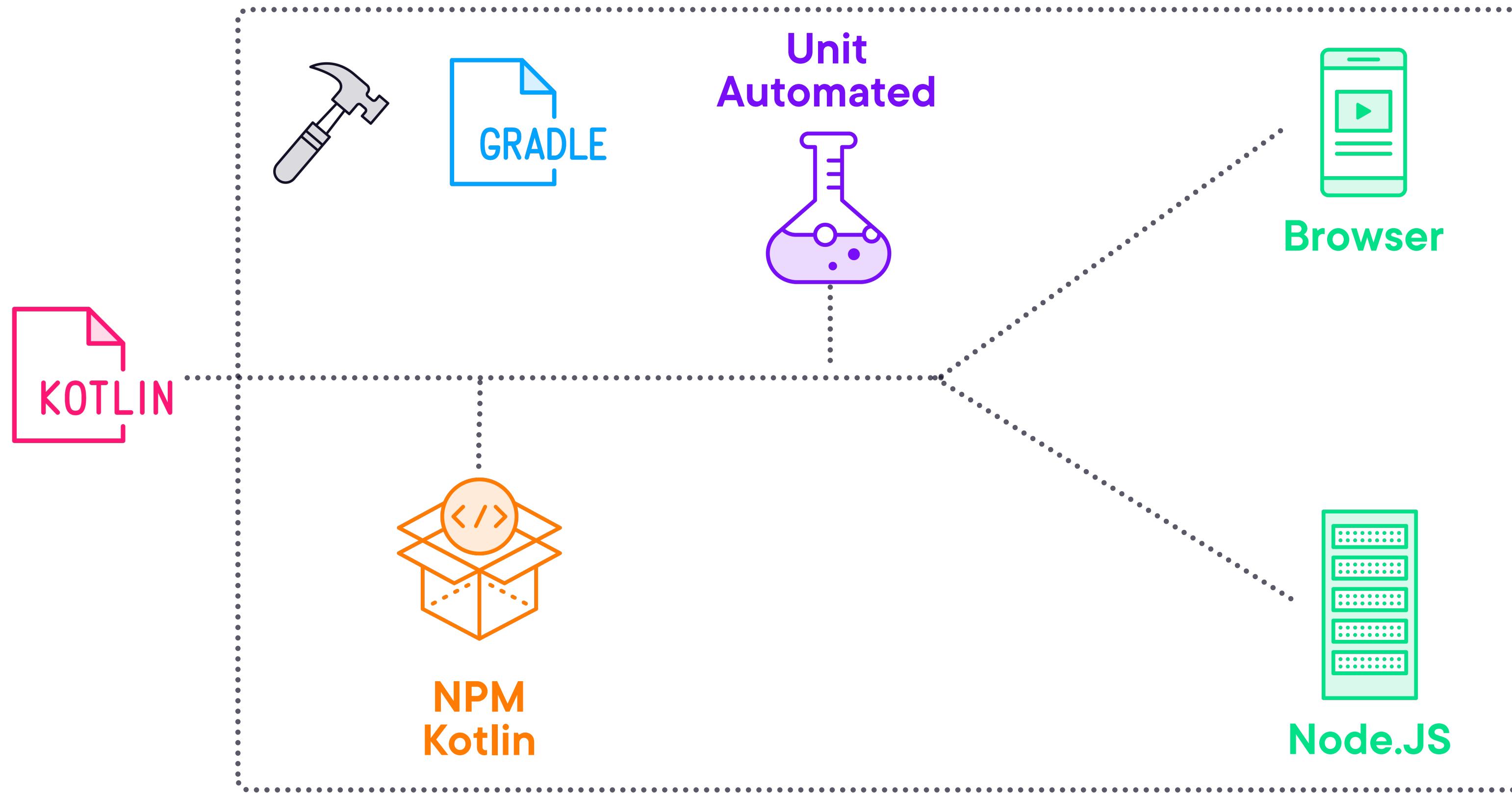
Browser

React

Node.JS



Implementation Path



Code Comparison

Header.js

```
function Header({ underline, text }) {  
  return <p style={underline ? "text-decoration: underline" : ""}>  
    {text}  
  </p>;  
}  
  
export default Header
```

Header.kt

```
external interface HeaderProps : Props{  
  var underline: Boolean  
  var text: String  
}  
val Header = FC<HeaderProps>{props ->  
  p {  
    if (props.underline){  
      css {textDecoration =  
        TextDecoration.underline}  
    }  
    +props.text  
  }  
}
```





Pros of Cross-platform Development

Cross-platform Benefits

**Write shared
code once**

**Language
standardization**

Easier debugging

Familiar language

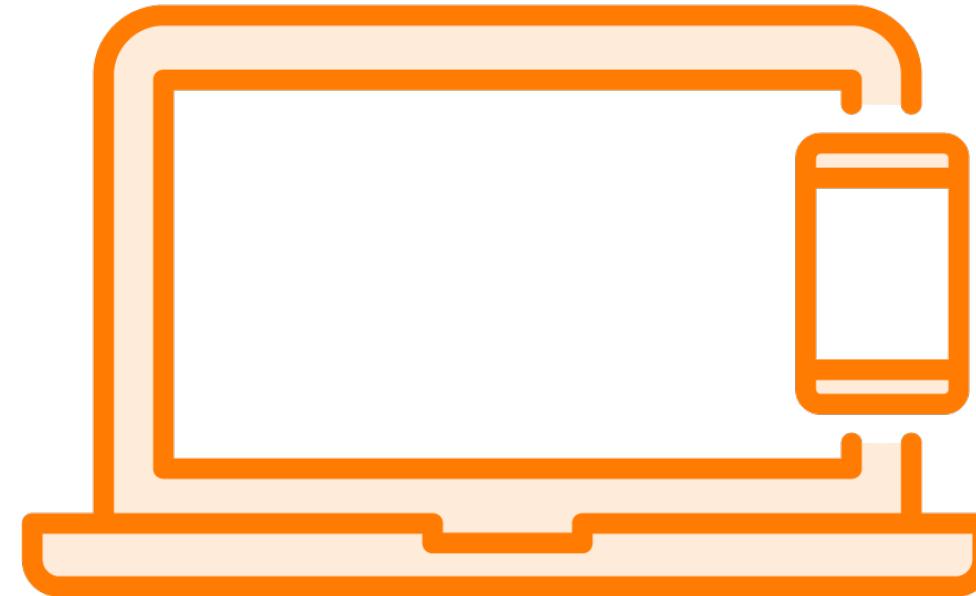
Native UIs



Cons of Cross-platform Development



Cons of Mobile Cross-platform Development



Native implementations are still required in some cases

Swift only dependencies aren't supported



Kotlin for JS Cons

TypeScript can provide strong types

Unnecessary complexity

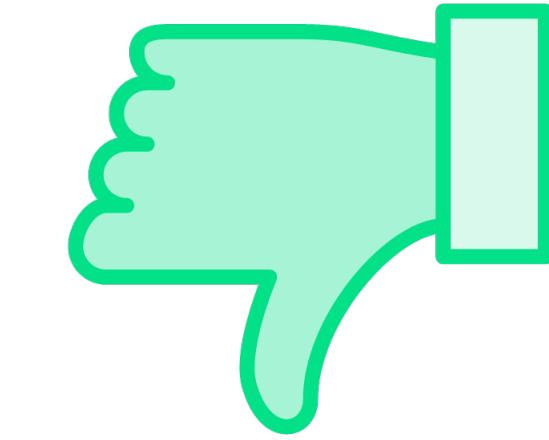
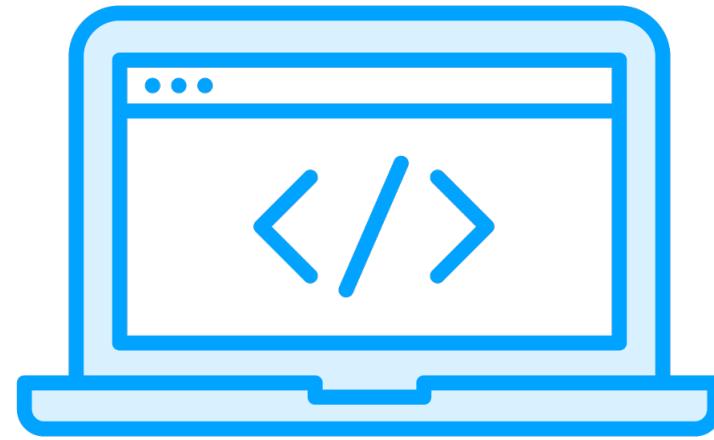
Low maturity

Limited applicability

Minimal shared code with mobile



General Cross-platform Concerns



**Engineer hesitation to
learn/maintain
another language**

**Broad Kotlin
disadvantages**





Replacing Java with Kotlin



**Kotlin's interoperability with
Java makes migrating
flexible and easy.**



Migration Approaches

Convert on a piece-by-piece basis

Write new features in Kotlin

Convert the entire app

Don't migrate feature(s)



Notable Differences

Strings

Collections

Nullability



Summary



Kotlin for Android

Cross-platform mobile

Kotlin for JavaScript

Cross-platform pros and cons

Kotlin as a Java replacement



Up Next:

Kotlin IDEs and Essential Libraries

