

Divers IT: Women in IT

Modern Android development and
the future of mobile development



with Android software developer Maryam Dashtirahmatyabadi

Agenda

About Me

Who am I?

Career path

Life of an Android Developer

What do I do daily?

Modern Android Development

Android Studio

Kotlin

Jetpack library

Compose

Architecture

Asynchronous

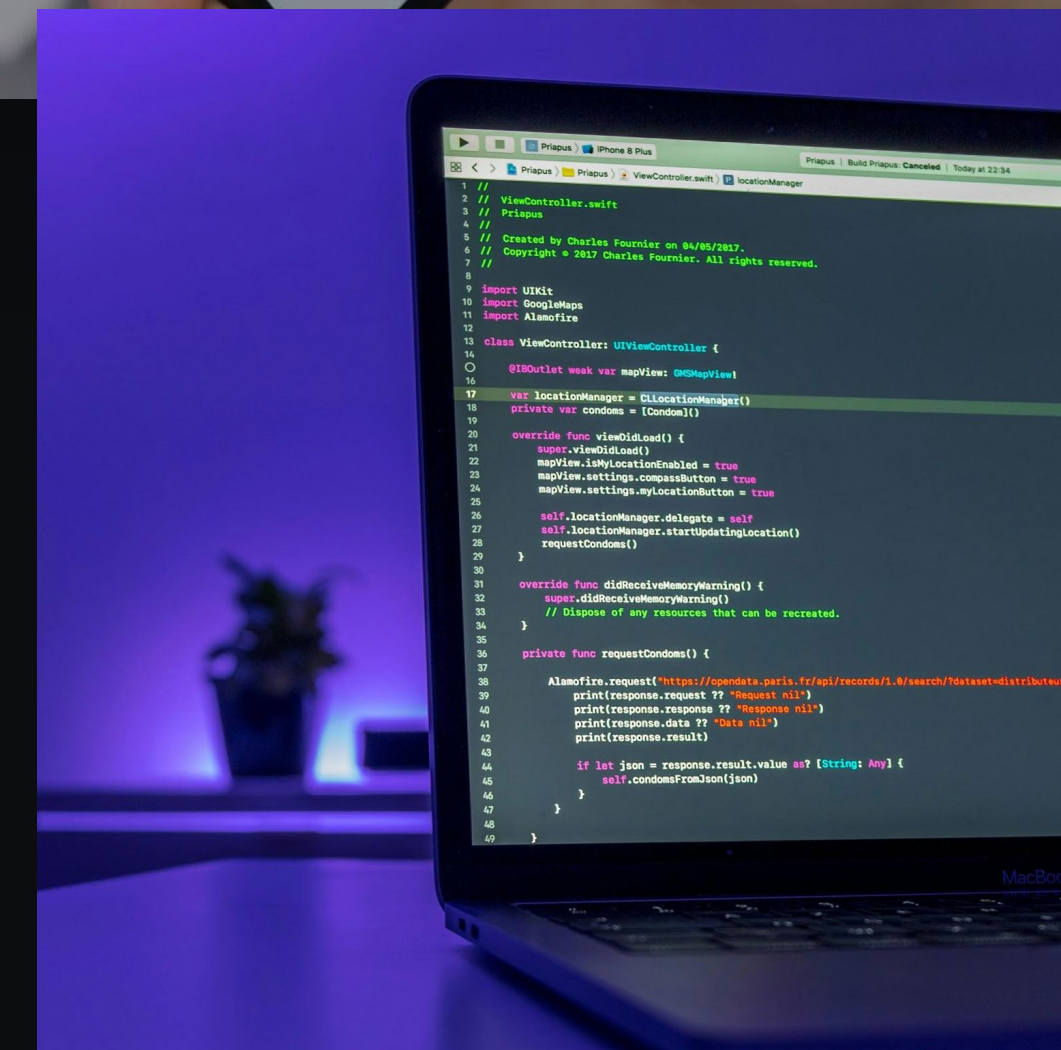
Android developer roadmap

Future of the Mobile Development

Kotlin Multiplatform?

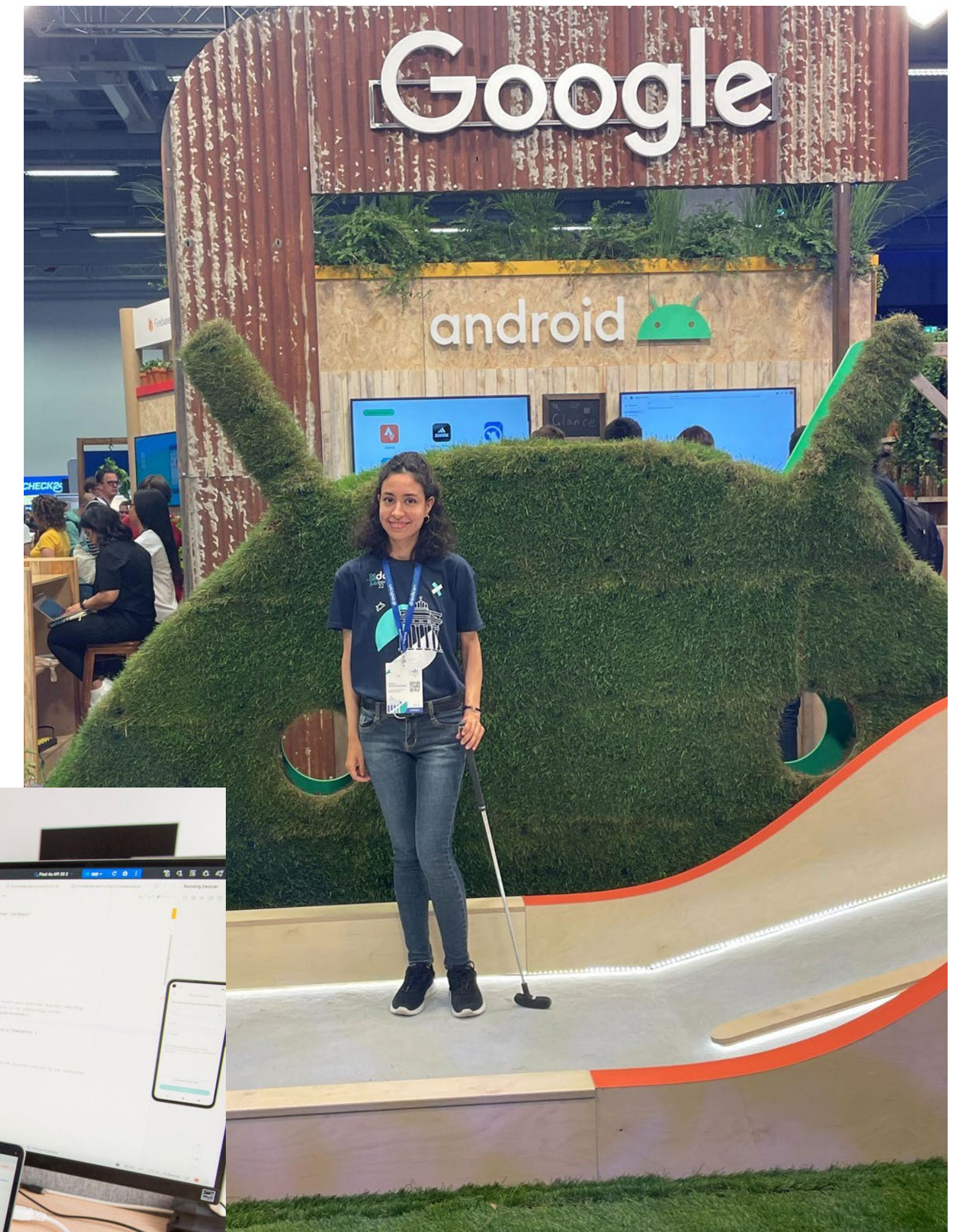
Application Demo

About Me



Who Am I?

- Maryam Dashtirahmatabadi
- Android Software Developer at Apps factory
- Moved to Germany at the end of Feb 2022
- German will be my fourth language :)



Career Path

1

Studies
Bachelor in Computer
Science

2

Freelance Android Developer
Self-learning about mobile
development in the field of
the Android platform

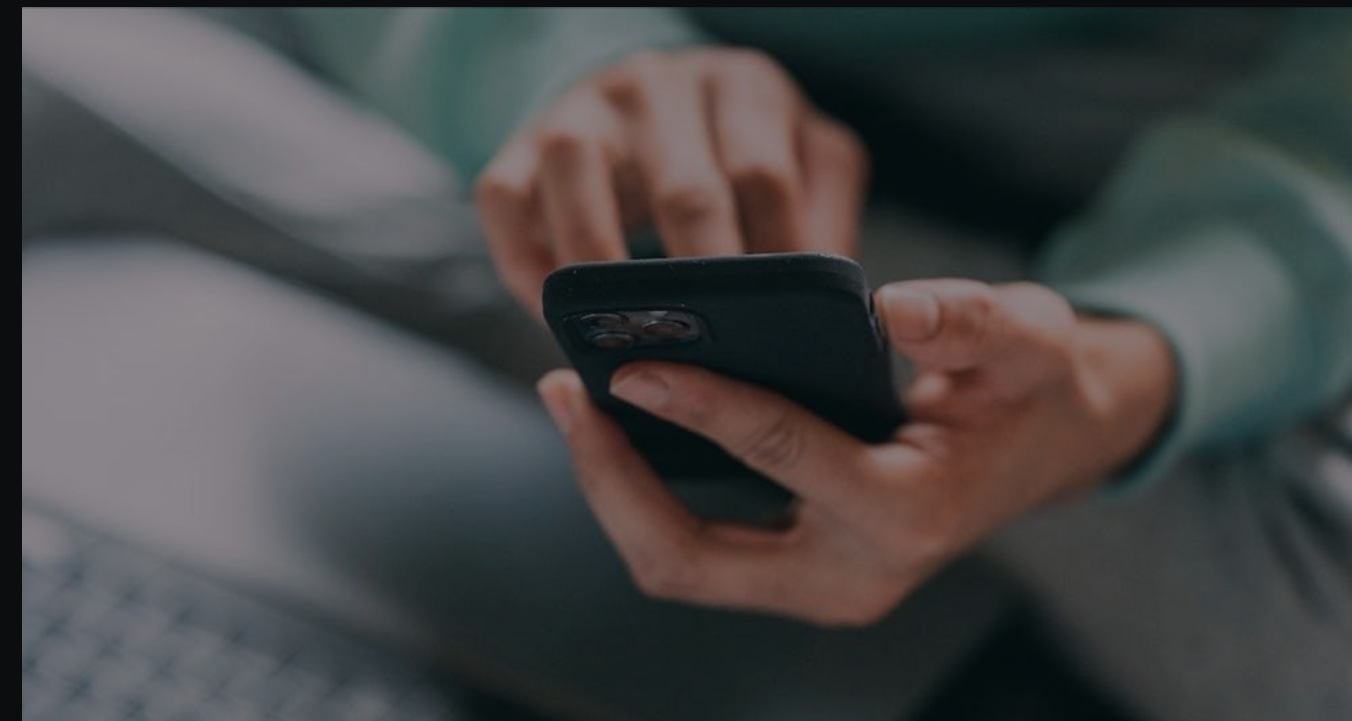
3

Full-time Android Developer in
Iran
Gain ~ 4.5 years of development
experience

4

Full-time Android Developer in
Germany
Since March 2022 continuing my
career in Germany

Life of an Android Developer



What do I do daily?

Programming



- Implementing new features
- Bug fixing
- Maintenance of the existing code
- Refactor the code

Learning



- Self-learning
- Workshops
- Tech conferences

Design Thinking



- Providing new ideas/features
- Improvement of existing features
- MVP planning

Dev Meetings



- Internal Android dev meetings
- Sync with the Backend team
- 1on1 with my tech lead

General Meetings



- App review
- Daily, Refinement, and Retro
- Onboarding new teammates

Sharing Knowledge



- Posting in internal dev channel
- Writing articles

Modern Android Development



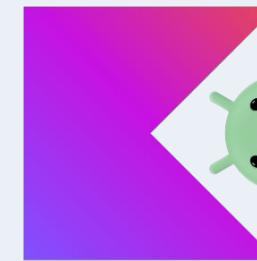
Tools

Modern Android Development

Android Studio



Kotlin



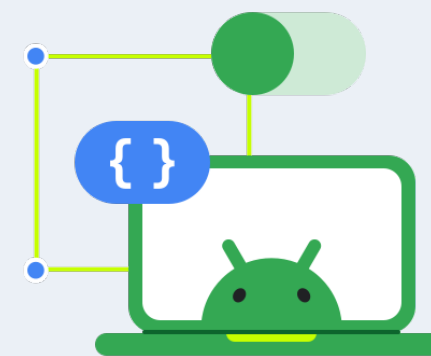
Compose



Jetpack Library



Architecture



Asynchronous



Android Studio Bot



How do I add dark theme support to my app?



What does Kotlin code look like?

```

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView {
            Greeting(name = "Android")
        }
    }
}

@Composable
fun Greeting(name: String) {
    Text("Hello $name!")
}

```

Nullability in the type system helps prevent NullPointerExceptions

Semicolons are optional

Lambdas allow you to pass code to a function as a parameter

Named parameters make code easier to read

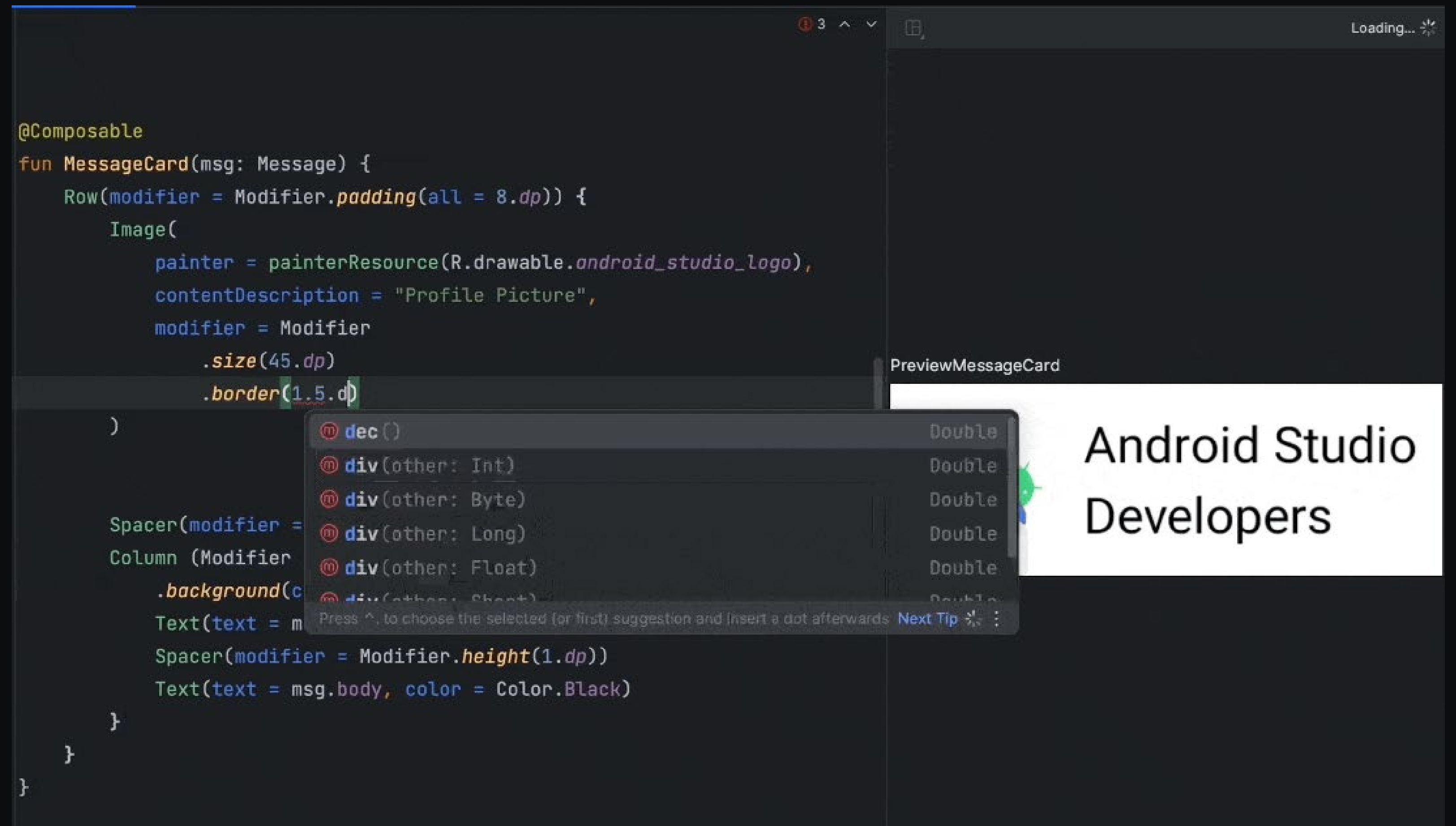
String templates simplify concatenation

Jetpack Library

- Collection of different libraries
- Provide best practices
- Reduce boilerplate code
- [Click here for more details](#)

<u>activity</u> *	Access composable APIs built on top of Activity.
<u>appcompat</u> *	Allows access to new APIs on older API versions of the platform (many using Material Design).
<u>appsearch</u> *	Build custom in-app search capabilities for your users.
<u>camera</u> *	Build mobile camera apps.
<u>compose</u> *	Define your UI programmatically with composable functions that describe its shape and data dependencies.
<u>databinding</u> *	Bind UI components in your layouts to data sources in your app using a declarative format.
<u>fragment</u> *	Segment your app into multiple, independent screens that are hosted within an Activity.

Jetpack Compose



Benefits of Architecture

- Improves the maintenance and quality of the application
- It allows the app to scale
- Easier to test
- It brings the consistency

Modern Android Architecture

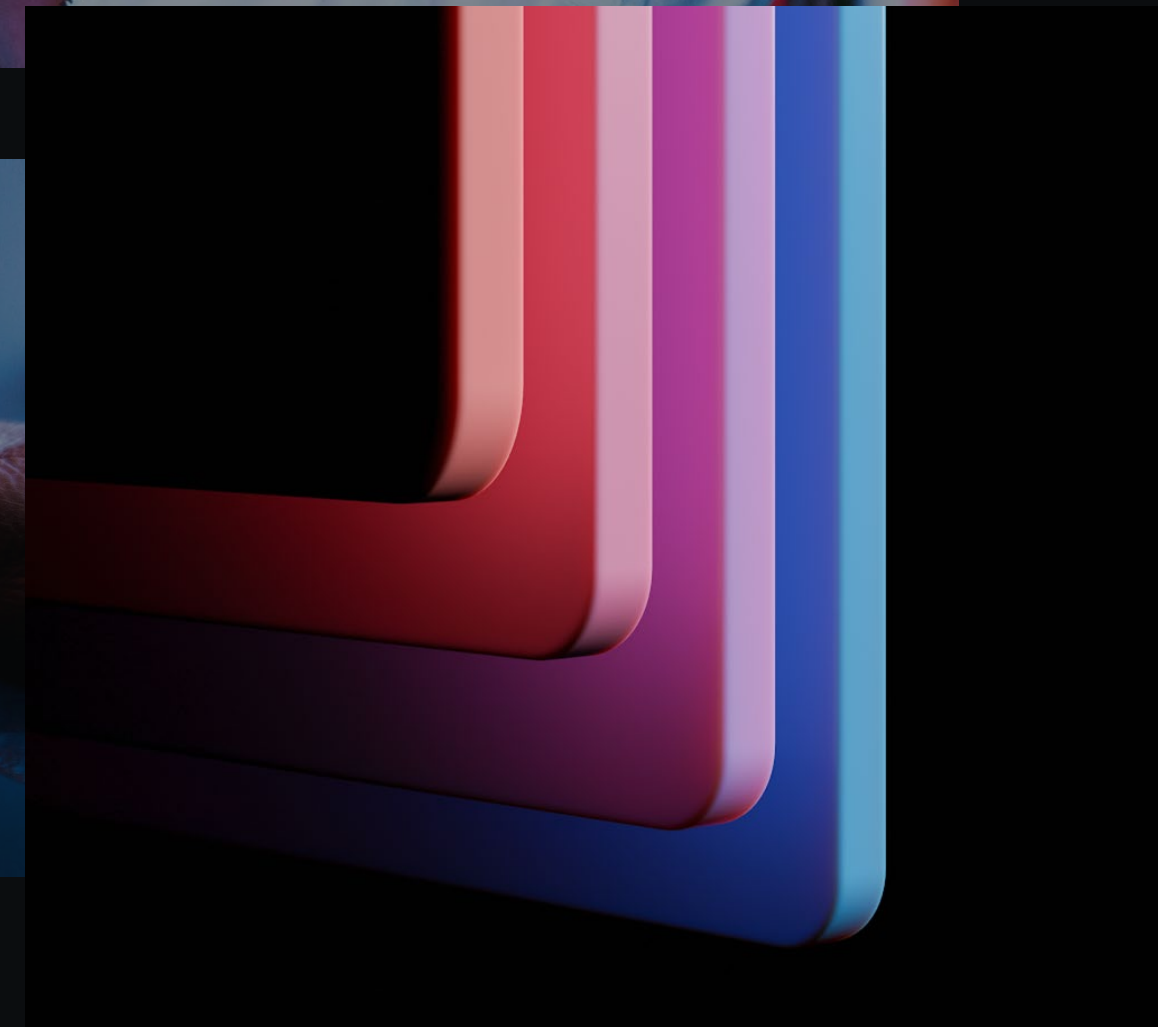
- A reactive and layered architecture
- Unidirectional data flow (UDF)
- A UI layer with state holders
- Dependency injection (DI)
- Coroutines and flows

Android Developer Roadmap

- Application fundamentals
- App Manifest
- Navigations
- Dependency injection
- CI/CD
- Design patterns
- Network
- Localstorage
- Service
- Code analysis
- Unit/UI tests
- App publishing
- ...
- [Click here for more details](#)

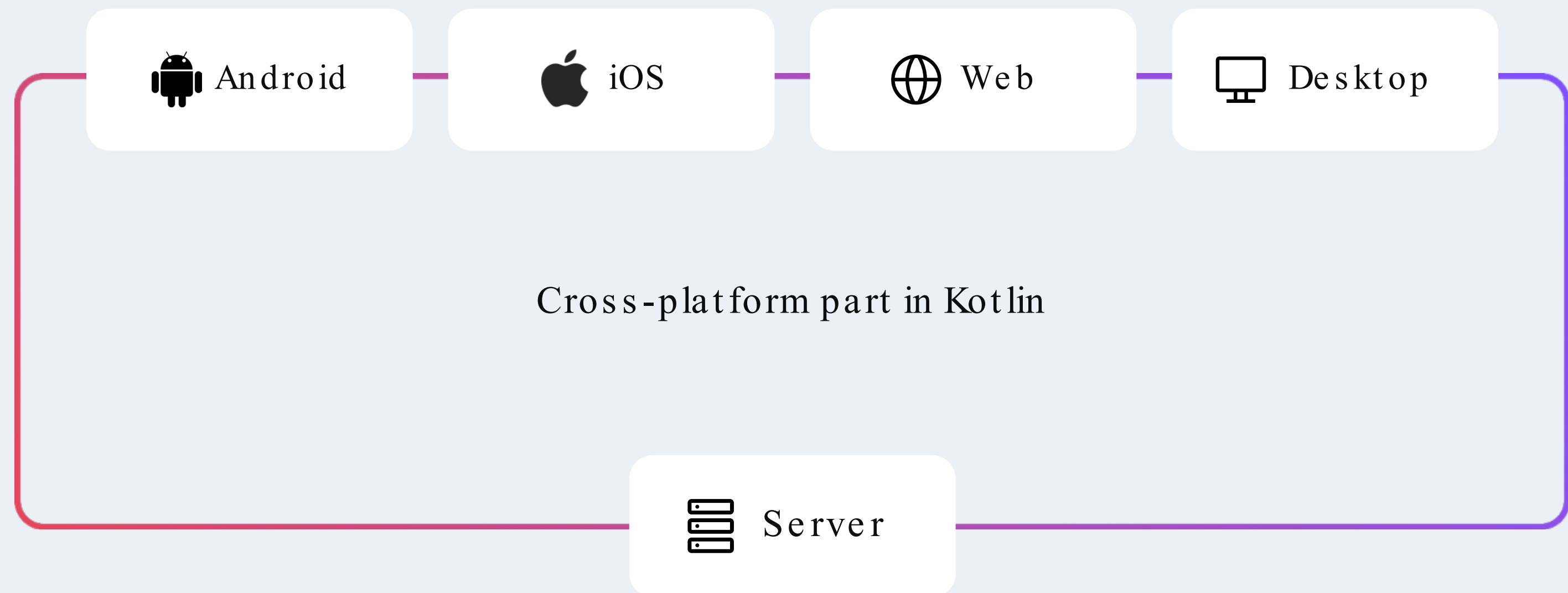


The future of Mobile Development



Ko t l i n M u l t i p l a t f o r m

- An open source technology by JetBrains
- Create applications for various platforms
- Simplify the development of cross-platform projects
- Reduces time spent writing and maintaining the code
- Retains the benefits of native programming
- Share code without compromising the quality



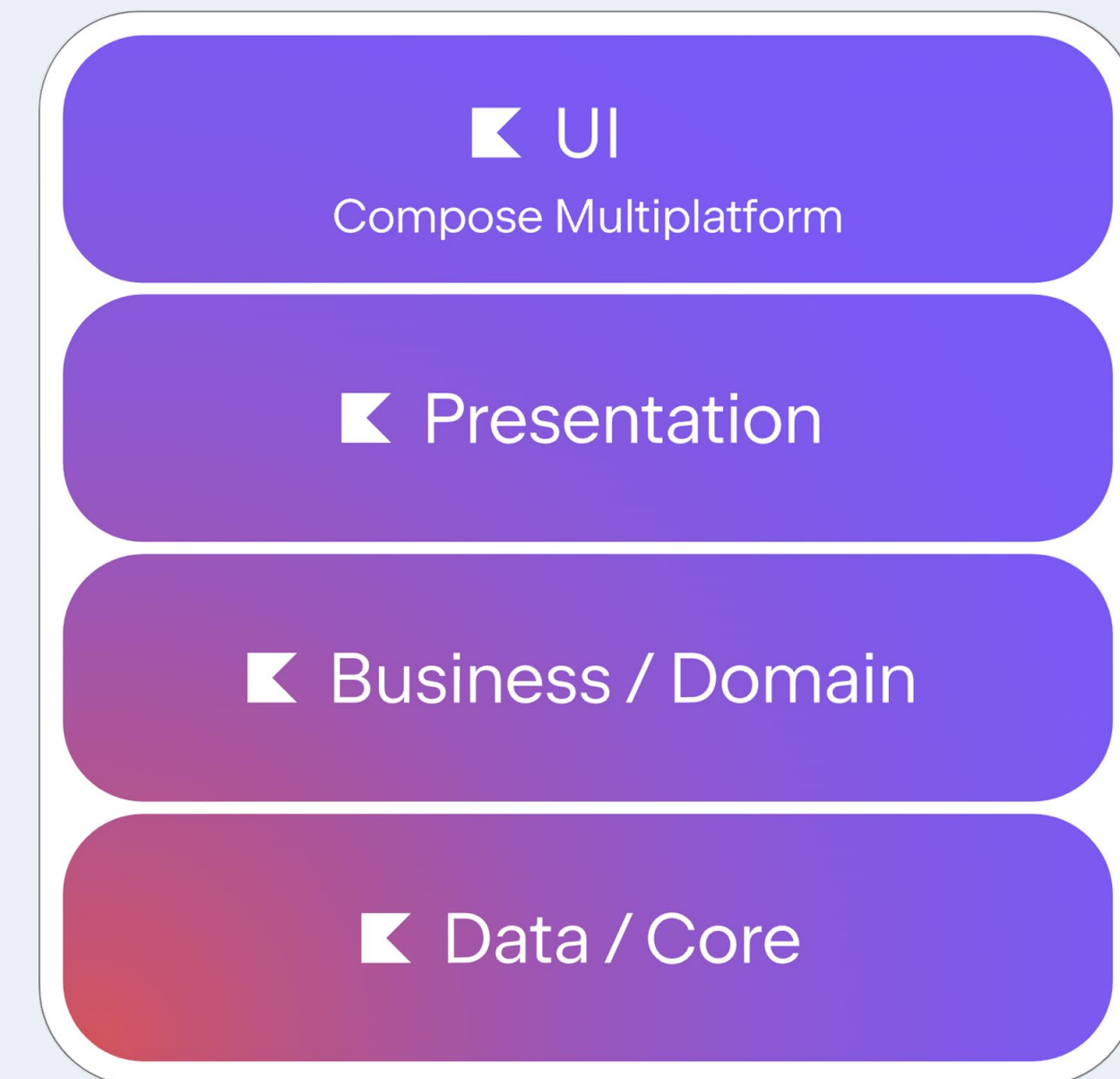
KMP

Kotlin Multiplatform Use Cases Android & iOS Applications

Sharing code between the mobile platforms is a major Kotlin Multiplatform use case.

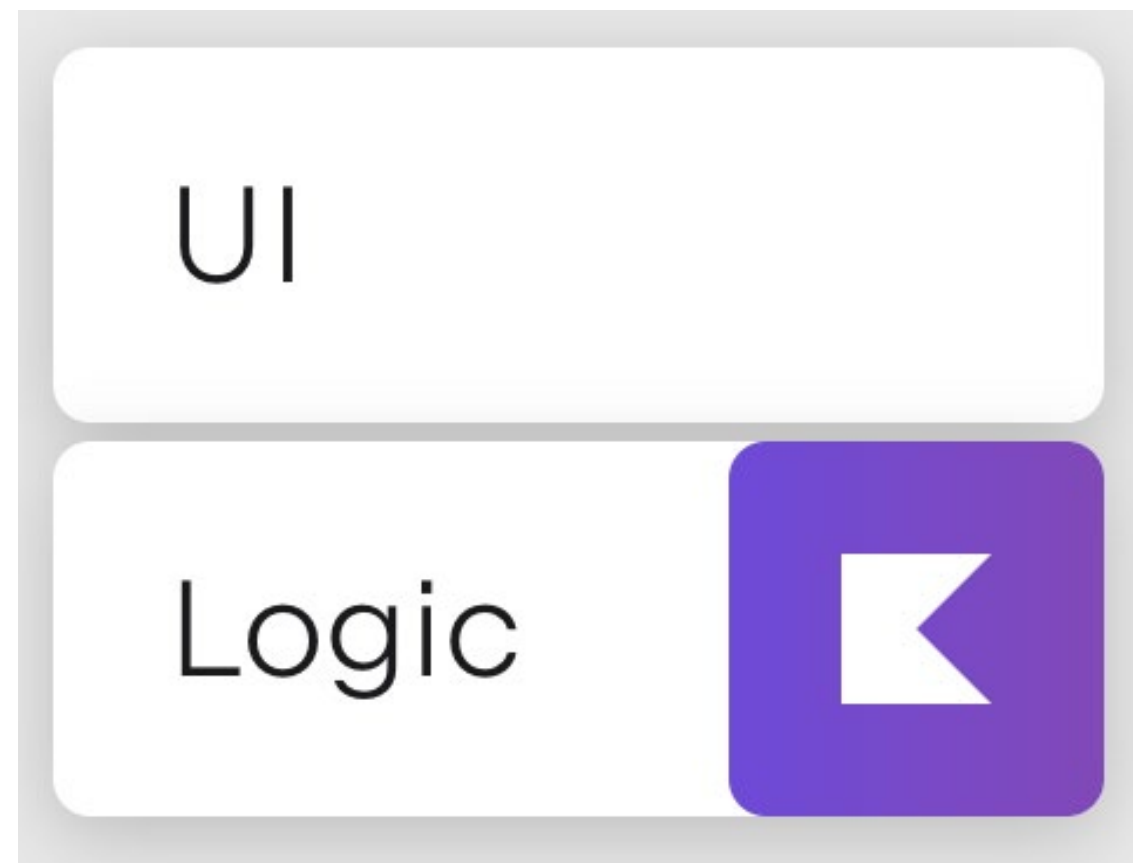
If you want to share the logic code between Android and iOS, you would need to use the recommended libraries provided by KMP instead of the native ones to implement the networking, data storage, and etc.

- **Ktor** : As an HTTP client for retrieving data over the internet.
- **kotlinx.serialization** : To deserialize JSON response into objects of entity classes.
- **kotlinx.coroutines** : Regarding asynchronous code
- **SQLDelight** : To generate Kotlin code from SQL queries and create a type-safe database API.
- ...

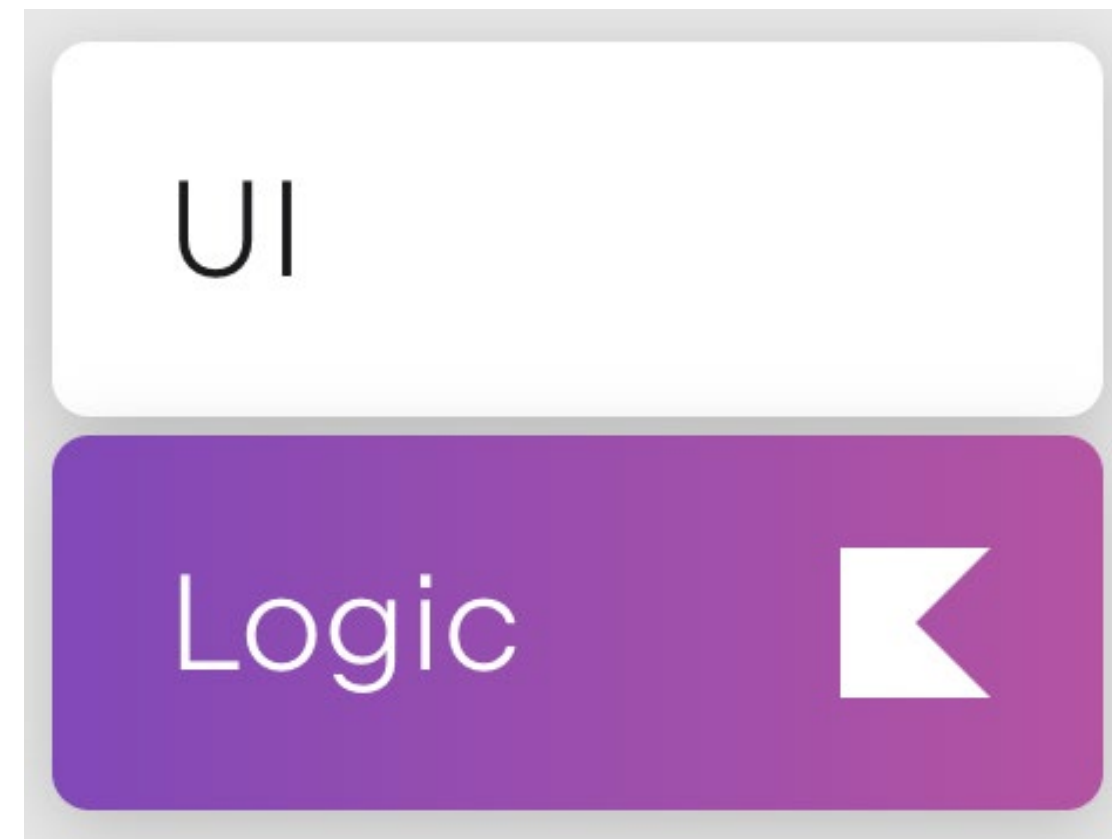


KMP

Suitable for All Kinds of Projects



Share a piece of logic

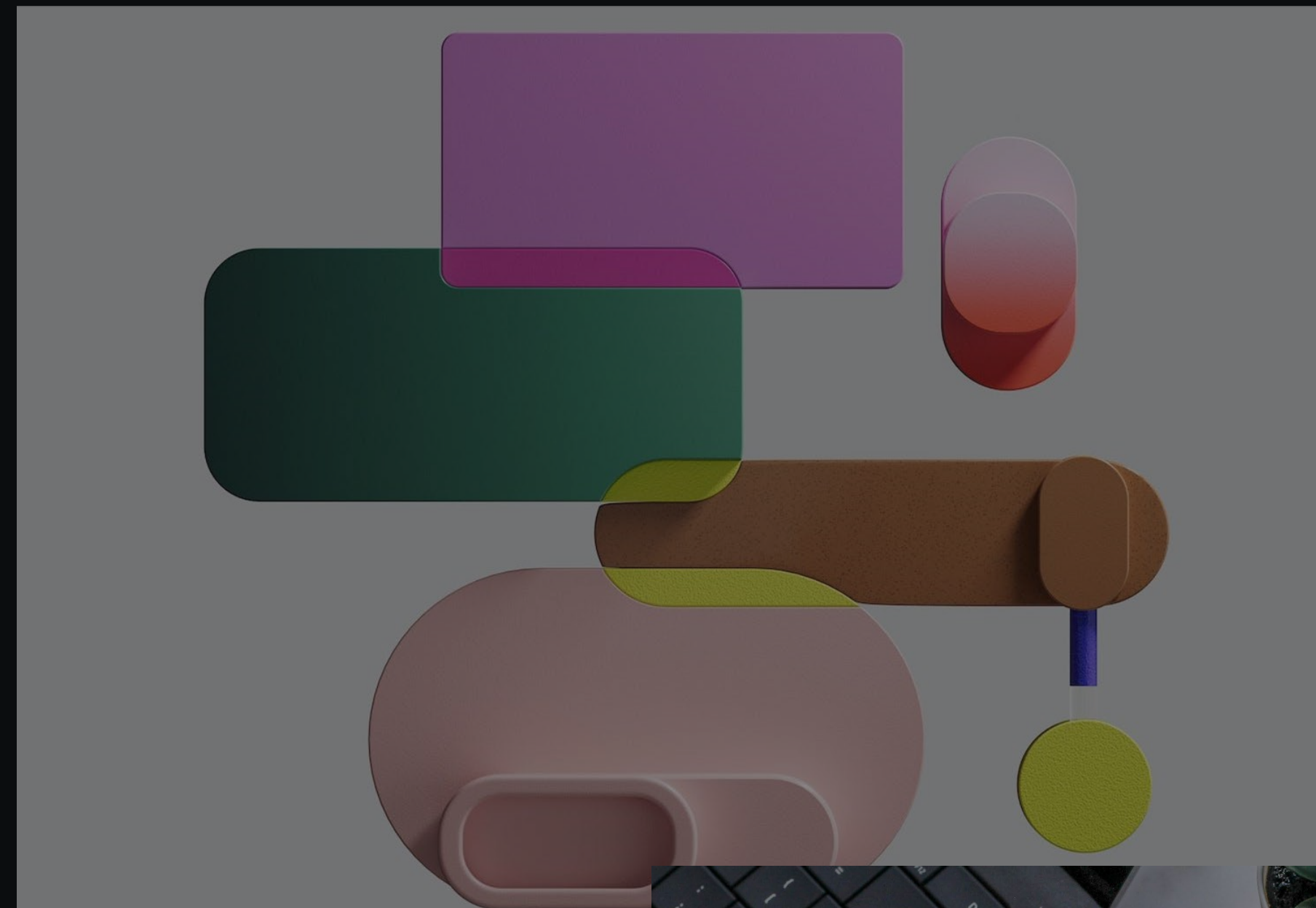


Share logic and keep UI native



Share up to 100% of the code

Application Demo





Thank You.



Maryam Dashtirahmatabadi

Android Software Developer

maryam.dashtirahmatabadi@appsfactory.de

Copyright ©

Dieses Werk ist geistiges Eigentum der Apps factory GmbH.

Es unterliegt einschließlich aller seiner Teile urheberrechtlichem Schutz. Jegliche Form der Verwertung, insbesondere jede Vervielfältigung, Verbreitung und Ausstellung sind, sofern sie nicht ausdrücklich vom Urheberrecht gestattet werden, verboten, es sei denn, die Appsfactory GmbH hat zuvor in die Verwertung schriftlich eingewilligt.

Die Verwertung sowie die versuchte Verwertung außerhalb der gesetzlich geregelten Fälle sind Straftaten nach §§ 106ff. UrhG. Verstöße können mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft werden.

Der Verletzer macht sich zudem schadensersatzpflichtig.

—

This document is protected by copyright. The use, or the distribution of this document without the explicit consent from Apps factory GmbH is forbidden. The content of this presentation is to be treated confidentially. Passing on content or information to third parties is not allowed.