

Modern app development

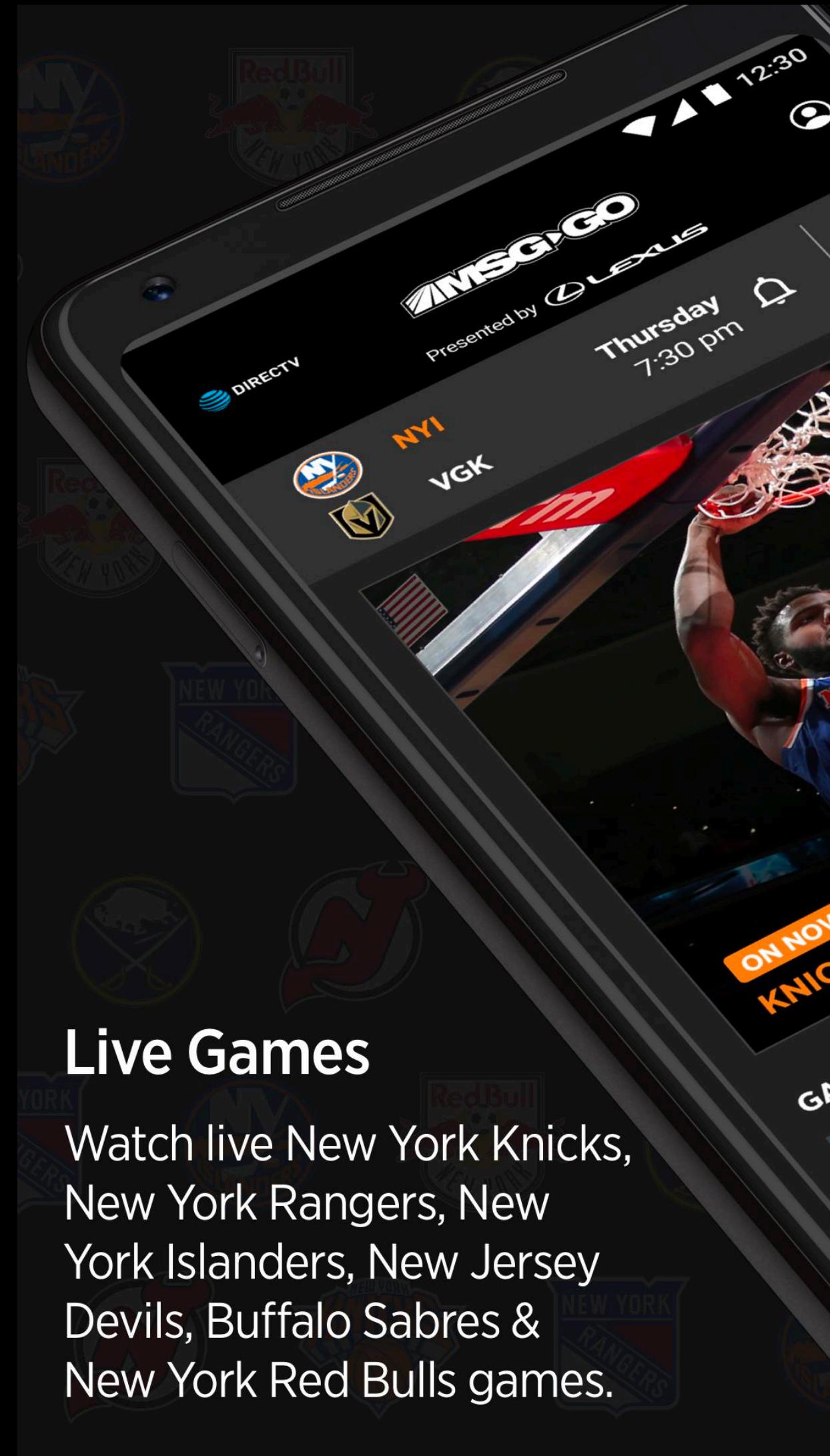
Norman Vicente

Who I am?

- Consultant at Xumak
- Android Software Developer for Wunderman Thompson
- Master's Degree Student in Data Science
- Past conferences: TedxGuatemala, Google I/O Extended Guatemala, Balanya University Club



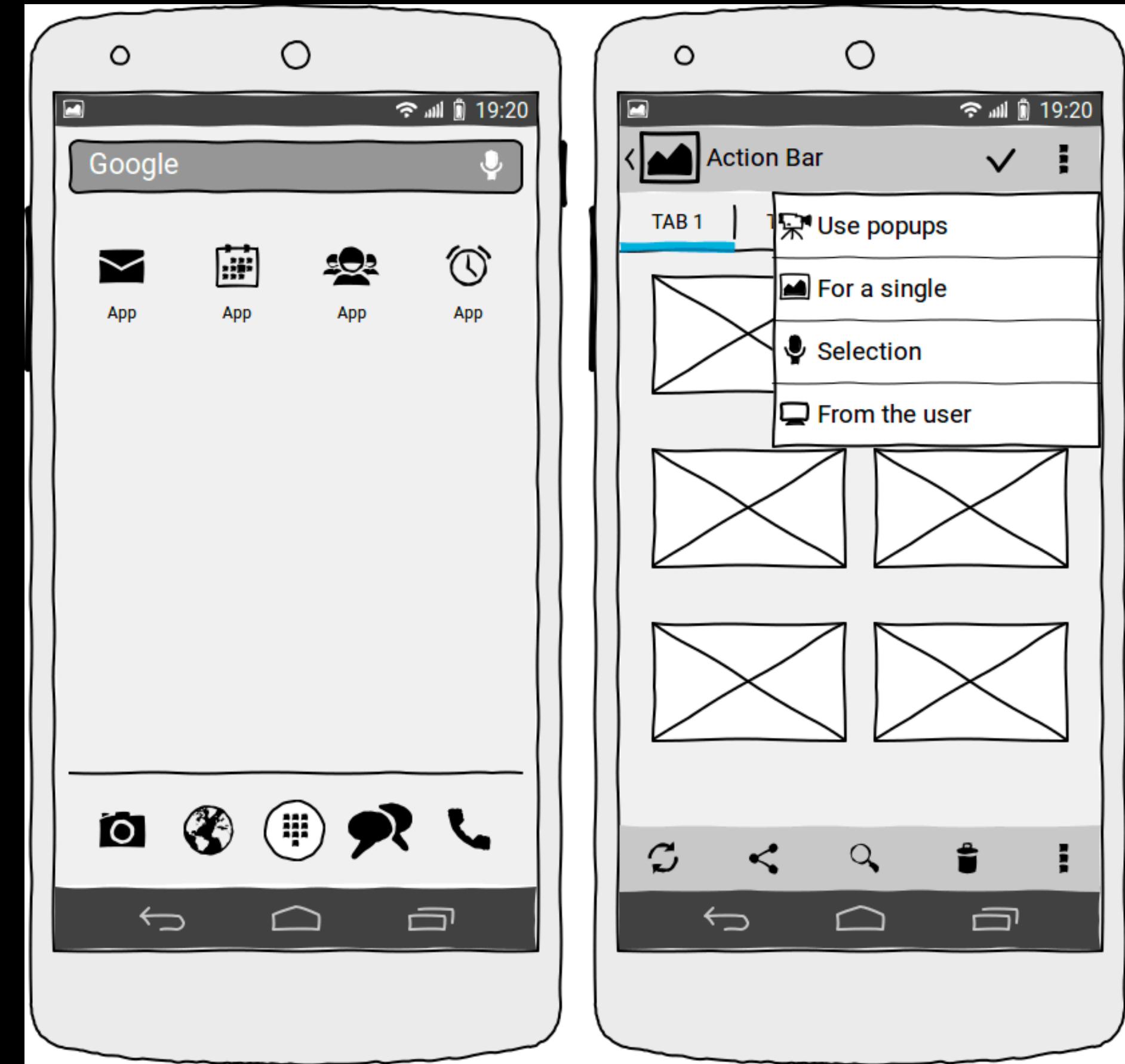
2021 Android Development roadmap



Live Games

Watch live New York Knicks,
New York Rangers, New
York Islanders, New Jersey
Devils, Buffalo Sabres &
New York Red Bulls games.

Mockup - UI

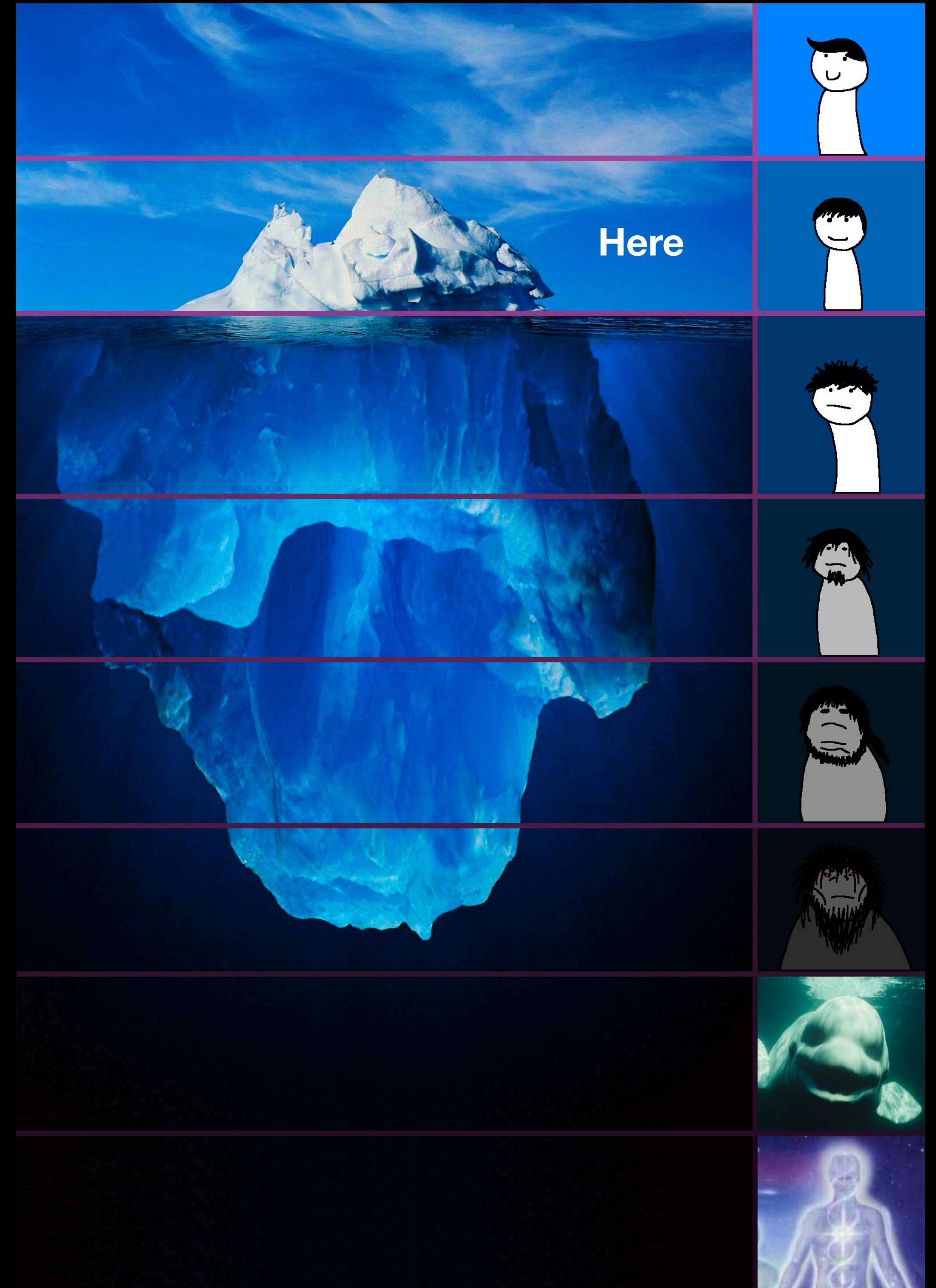


Programming Languages

- Native: Kotlin, Java
- Hybrid: Dart (Flutter), Javascript (React Native)

IDE - Text Editor





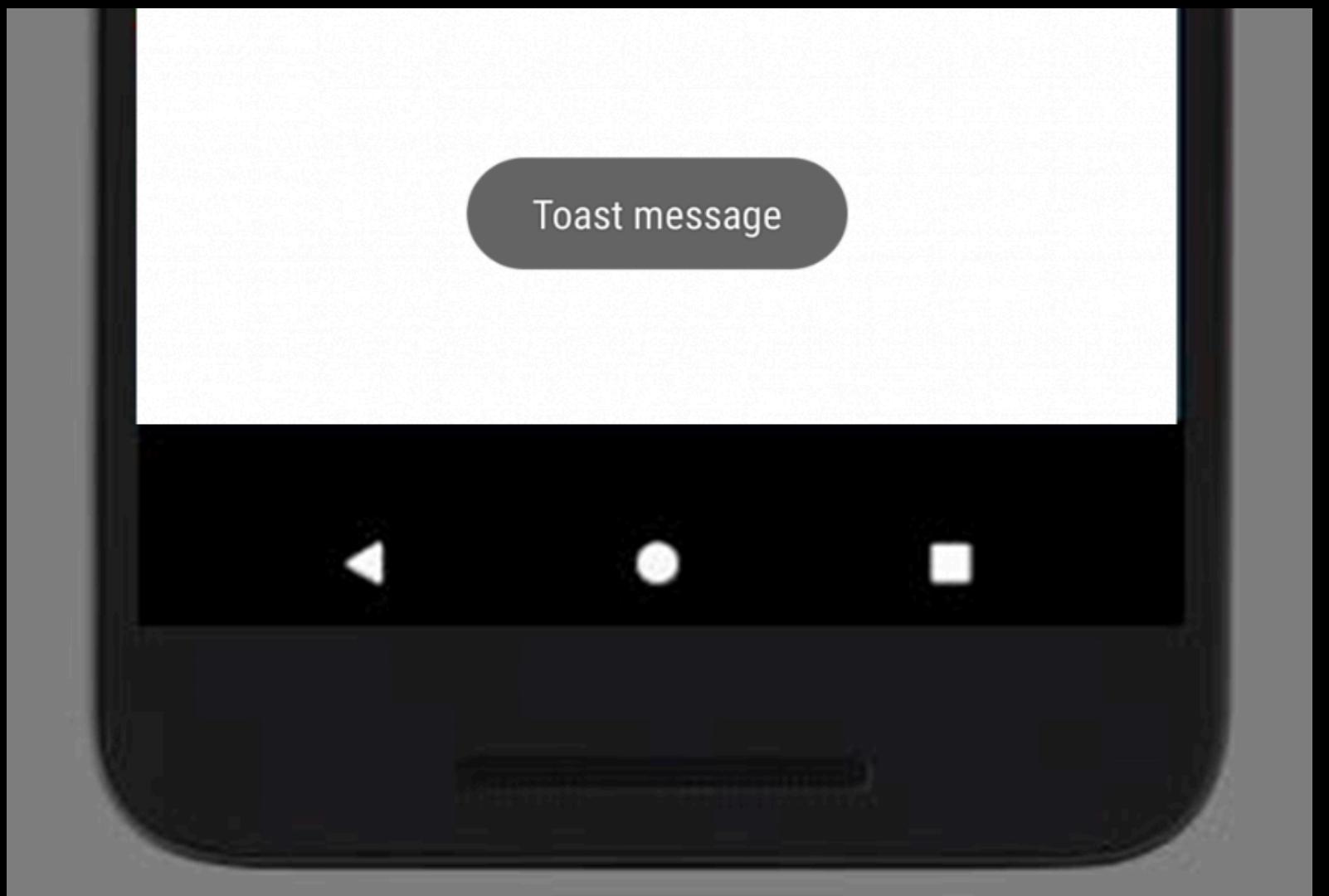
“toy example”

```
● ○ ●  
{  
    Toast.makeText(getApplicationContext(),"a toy example",Toast.LENGTH_SHORT).show()  
}
```

Java Example

```
● ○ ●  
{  
    Toast.makeText(applicationContext,"a toy example",Toast.LENGTH_SHORT).show()  
}
```

Kotlin Example





POJO



```
public class Person {  
    private String name;  
  
    public Person(String name){  
        this.name = name;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    // toString...  
    // hashCode...  
    // equals...  
    // copy...  
}
```

```
data class Person(val name: String)
```

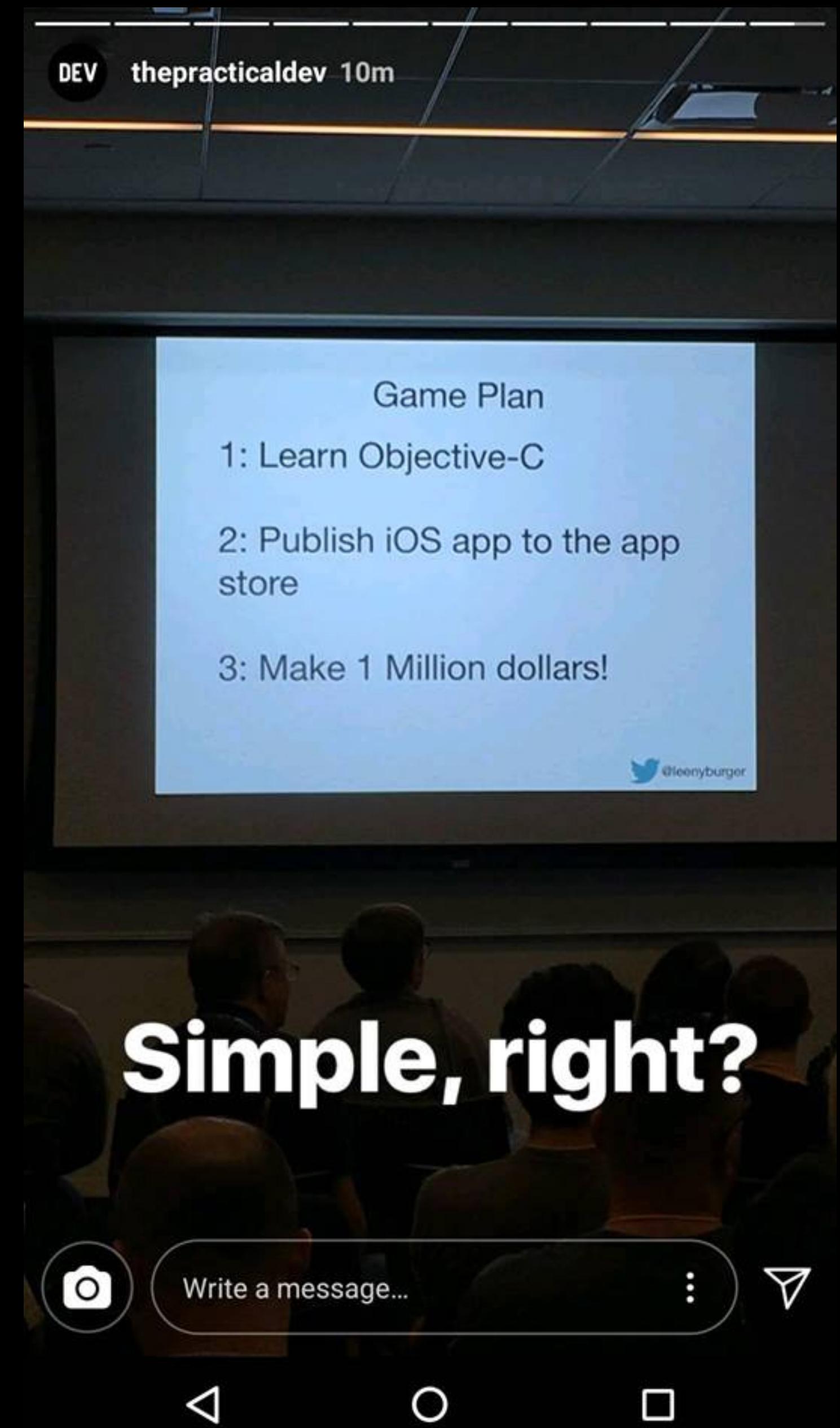


Code



```
public void createAndPrintPerson() {  
    String name = "Pieter";  
    Person person = new  
    Person(name);  
  
    printName(person.getName());  
    // Prints: Pieter  
}
```

```
fun createAndPrintPerson() {  
    val name = "Pieter"  
    val person = Person(name)  
  
    printName(person.name)  
    // Prints: Pieter  
}
```



Android app architecture

- Mockup - UI
- Programming language
- Android pattern
- Android documentation

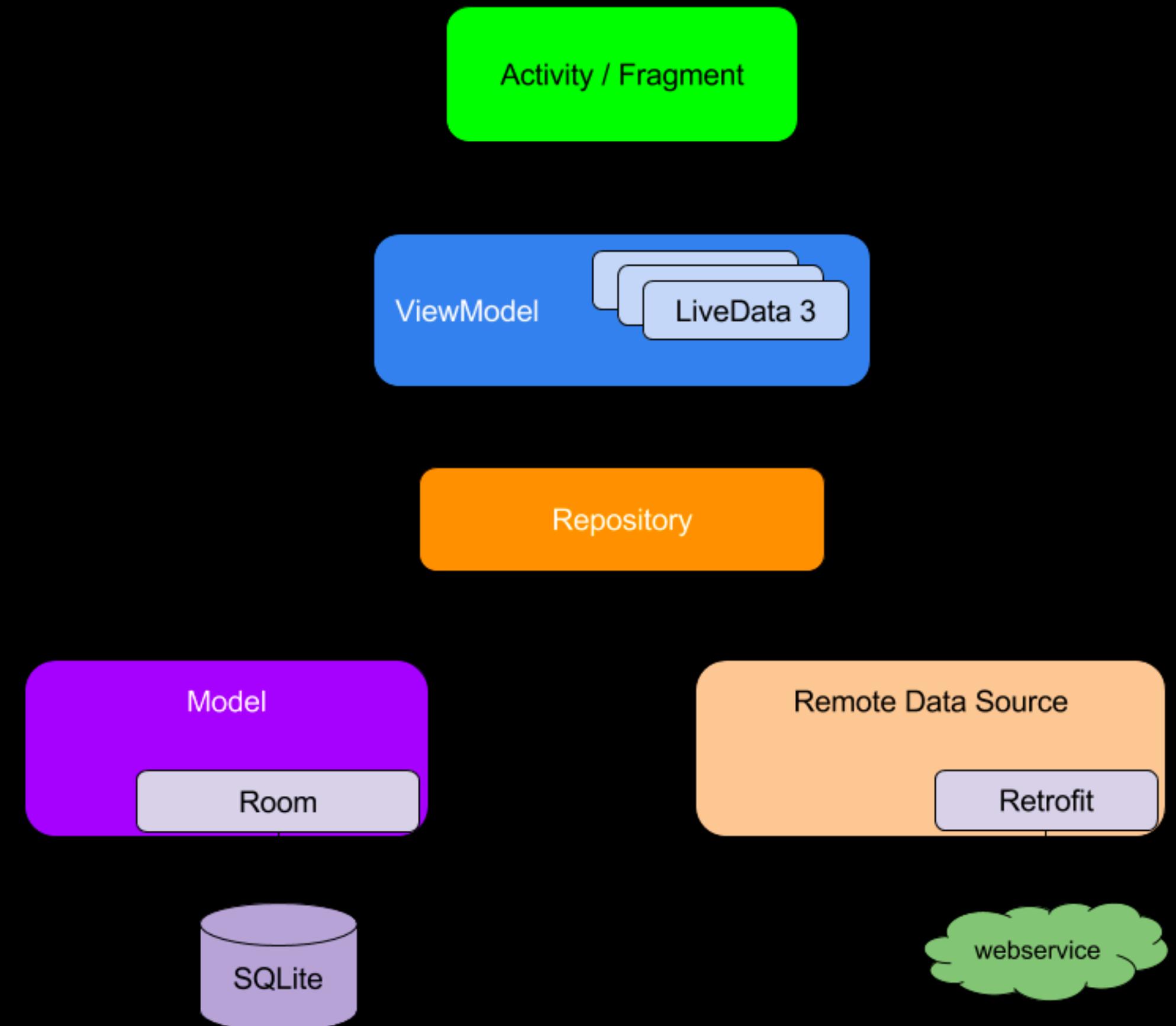
Android app architecture

- Mockup - UI
- Programming language
- Android pattern <--
- Android documentation

Common Architectural Patterns

- MVVM
- MVP
- MVP
- MVI

<https://developer.android.com/jetpack/guide>



Common libraries for modern android app

- *Image Handler*
 - Glide, Picasso, Coli
- *Networking*
 - Retrofit, OkHttp
- *Dependency injection*
 - Dagger, Hilt

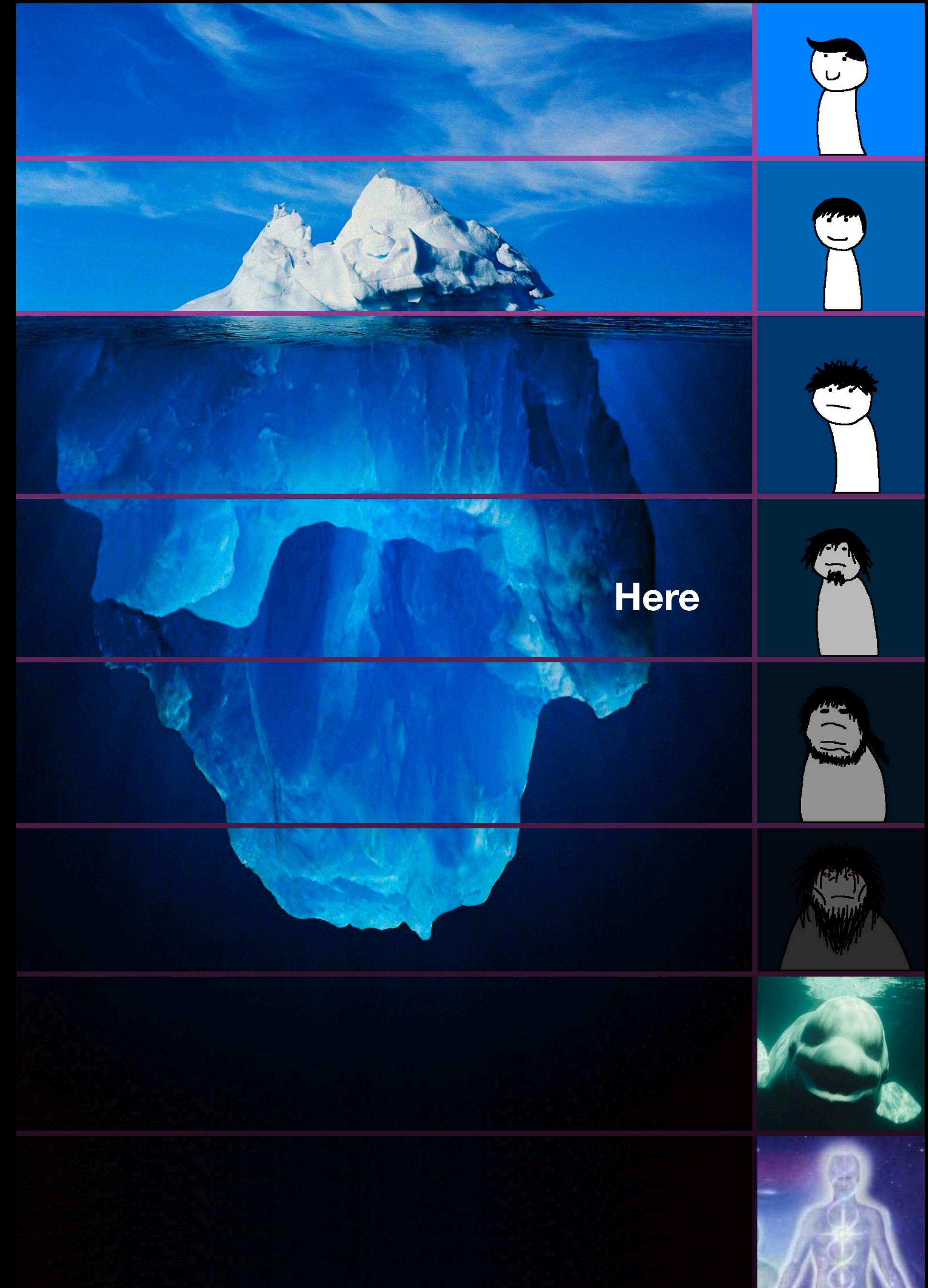
Android documentation

- Kdoc
- JavaDoc



A screenshot of an Android application window showing Java code with multi-line comments. The code defines a class `MovieViewModel` with a constructor parameter `app: Application`. It contains a private variable `movieLiveData: MutableLiveData<Movie>` and a function `getGreeting(name: String)`.

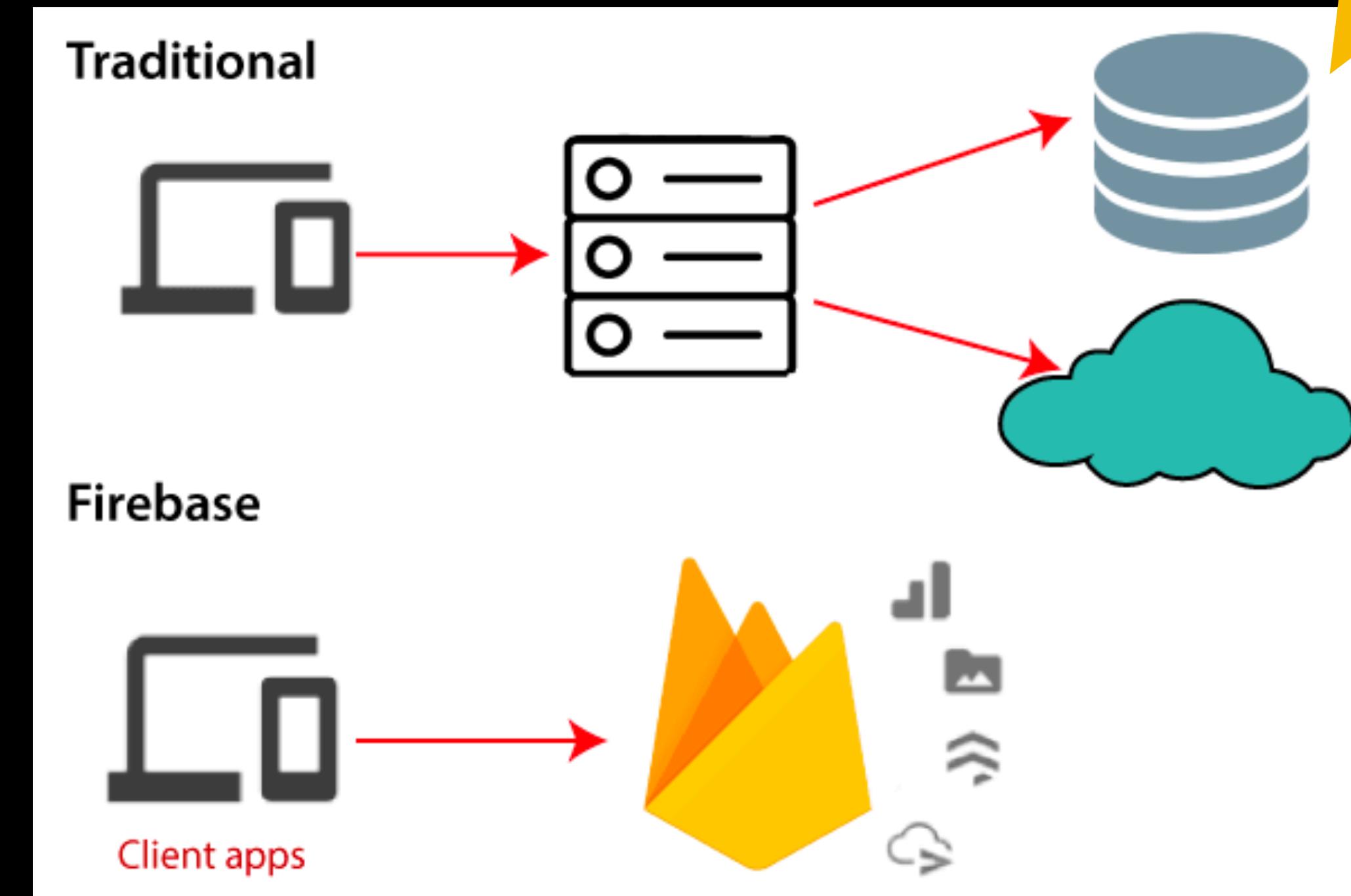
```
/**  
 * Example of a class comment.  
 *  
 * @param app: Example of a param comment.  
 */  
class MovieViewModel (app: Application ): AndroidViewModel{  
  
    /**  
     * Example of a property comment.  
     */  
    private val movieLiveData: MutableLiveData<Movie>()  
  
    /**  
     * Example of a function comment.  
     * @param name Example of param comment.  
     * @return Application: Example of a return comment.  
     */  
    fun getGreeting(name: String) = "Hello $name"  
}
```





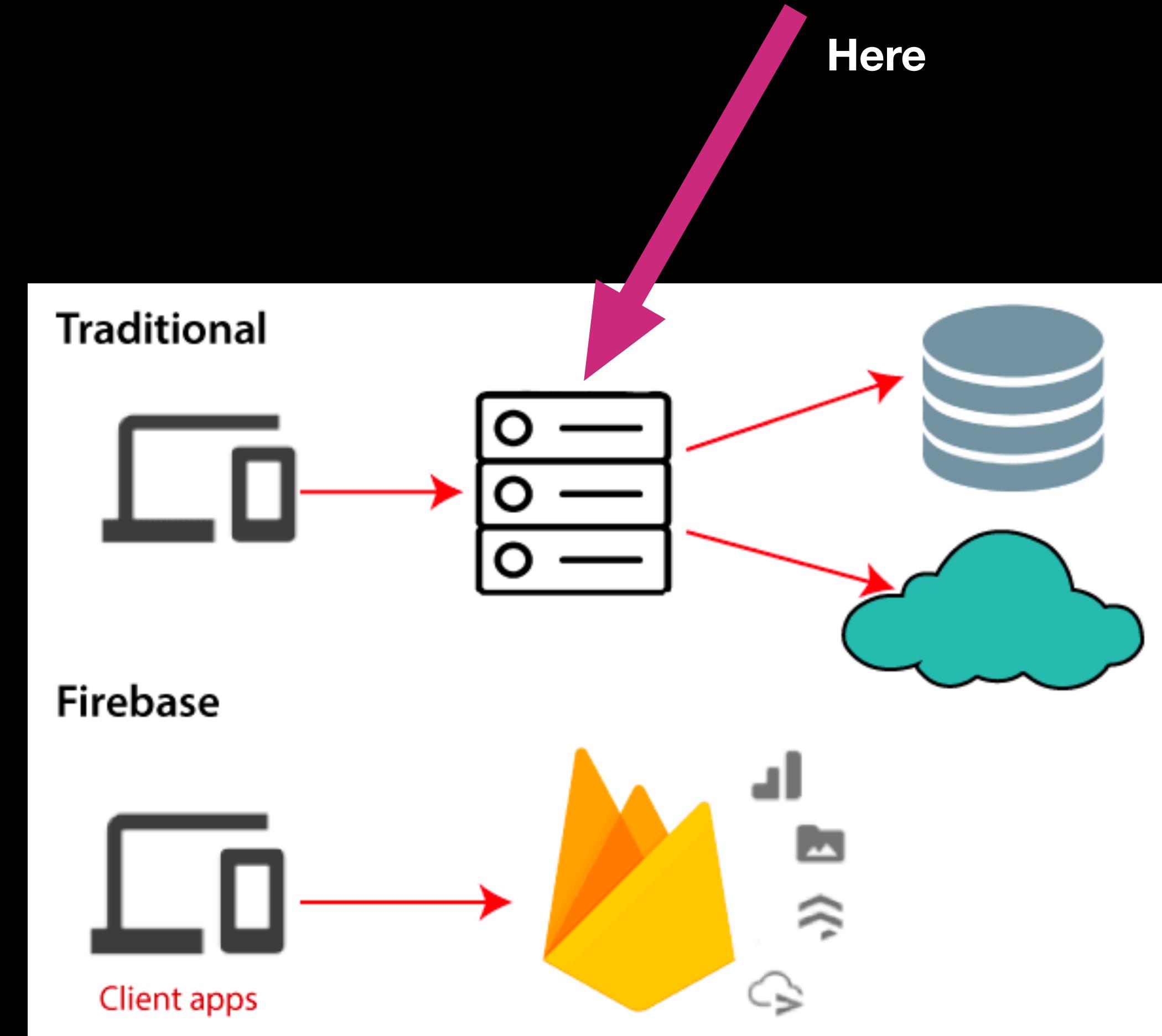
Outside of android
development

Now Here



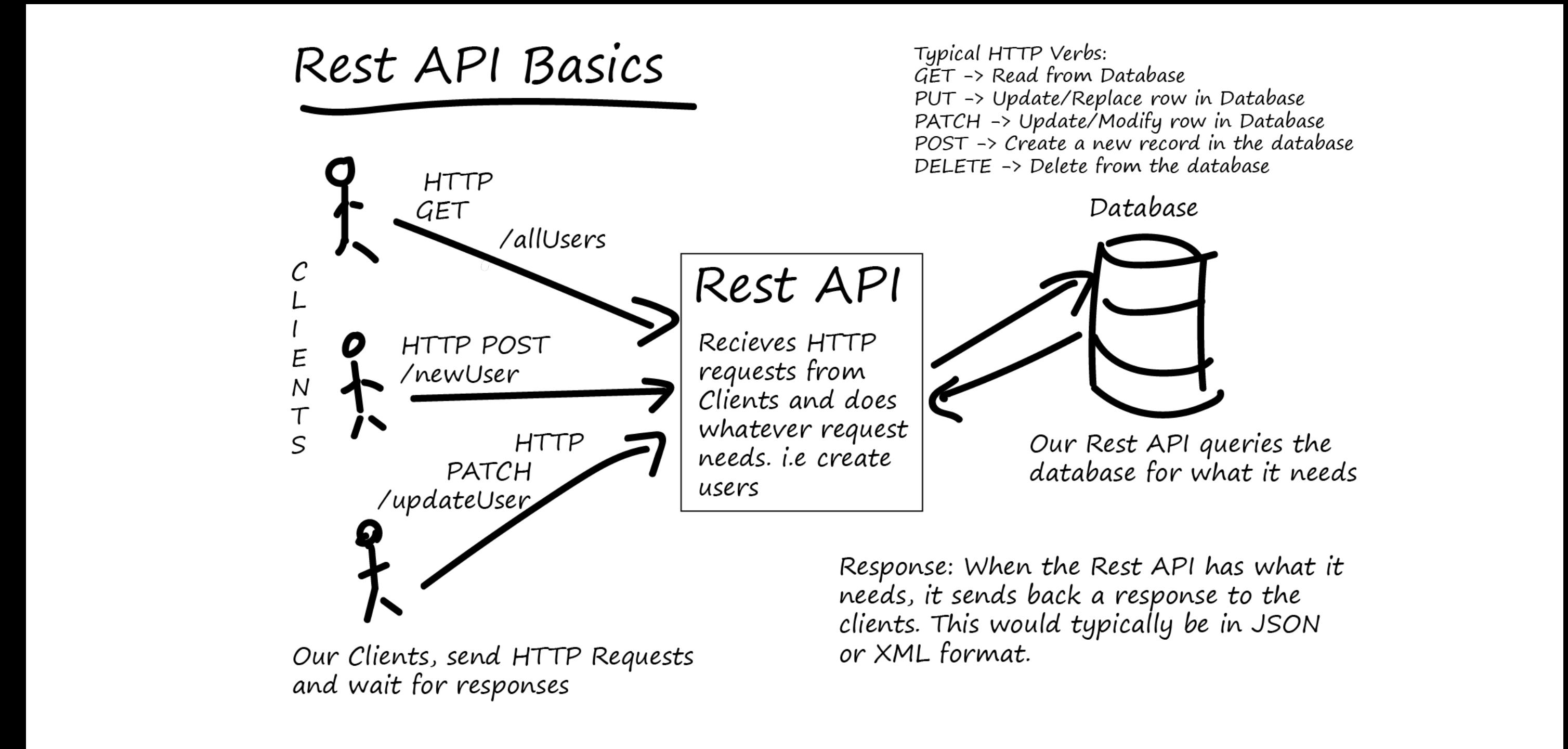
Data

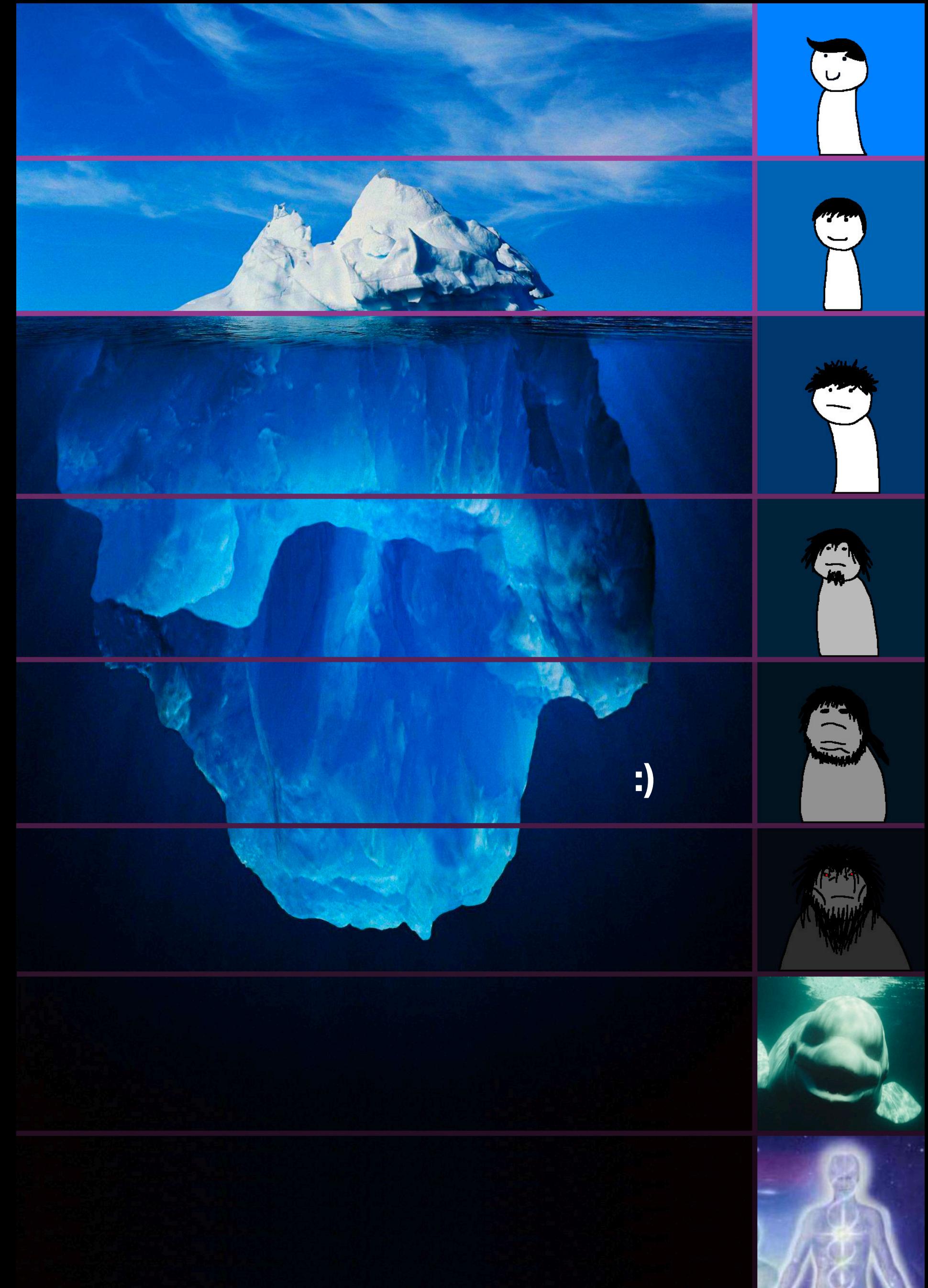
- Firebase
- MySQL
- SQL Server
- mongoDB



Getting data

- Firebase
- REST APIs



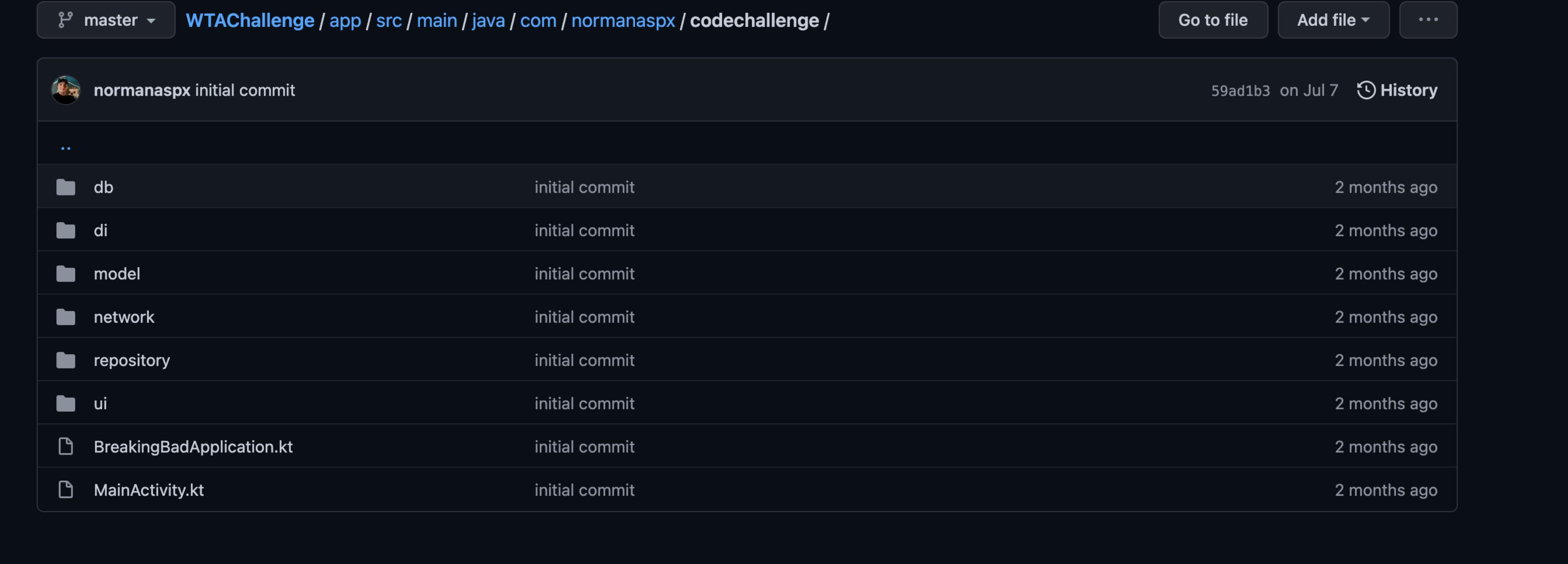


Some code for you



<https://github.com/normanasp/WTAChallenge>

Some code for you



A screenshot of a GitHub repository's commit history. The repository path is `WTAChallenge / app / src / main / java / com / normanaspX / codechallenge /`. The branch is `master`. The commit was made by `normanaspX` on July 7, 2018, with the commit hash `59ad1b3`. The commit message is `initial commit`. The commit details show the creation of several initial files and folders: `db`, `di`, `model`, `network`, `repository`, `ui`, `BreakingBadApplication.kt`, and `MainActivity.kt`. All files were committed 2 months ago.

File/Folder	Commit Message	Time Ago
<code>db</code>	initial commit	2 months ago
<code>di</code>	initial commit	2 months ago
<code>model</code>	initial commit	2 months ago
<code>network</code>	initial commit	2 months ago
<code>repository</code>	initial commit	2 months ago
<code>ui</code>	initial commit	2 months ago
<code>BreakingBadApplication.kt</code>	initial commit	2 months ago
<code>MainActivity.kt</code>	initial commit	2 months ago

<https://github.com/normanaspX/WTAChallenge>

Some code for you

```
19
20  /**
21   * This component is used for general purposes
22   * it is placed in the application component since
23   * it can be injected on any other modules where
24   * it is needed
25   *
26   * @author Norman Vicente
27   *
28   */
29  @Module
30  @InstallIn(ApplicationComponent::class)
31  object AppModule {
32
33
34      @Provides
35      @Singleton
36      fun provideRetrofit(): Retrofit =
37          Retrofit.Builder()
38              .baseUrl(BreakingBadApplication.API_BASE_URL)
39              .addConverterFactory(MoshiConverterFactory.create())
40              .build()
41
42      @Singleton
43      @Provides
44      fun provideBreakingBadService(retrofit: Retrofit): BreakingBadService =
45          retrofit.create(BreakingBadService::class.java)
46
47      @Provides
48      fun provideDao(db: BBDatabase) = db.characterDao()
49
50      @Singleton
51      @Provides
52      fun provideRoomInstance(@ApplicationContext ctx: Context) = Room.databaseBuilder(ctx, BBDatabase::class.java,"bb").build()
53
54      @Provides
55      fun providesCharacterRepository(characterDao: CharacterDao, service: BreakingBadService): CharacterRepository
56      = CharacterRepository(characterDao, service)
57
58
59 }
```

Some code for you

```
10  /**
11  * Created by Norman 7/7/2021
12  */
13
14  @Parcelize
15  @Entity(tableName="character")
16  data class Character (
17      @PrimaryKey
18      @SerializedName("char_id")
19      val char_id: Int,
20      var name: String,
21      val occupation: Array<String>,
22      val img: String?,
23      val status: String,
24      val nickname: String,
25      var isLiked: Boolean = false
26  ) : Parcelable
```

Some code for you

<https://github.com/normanaspX/WTAChallenge>

Some code for you

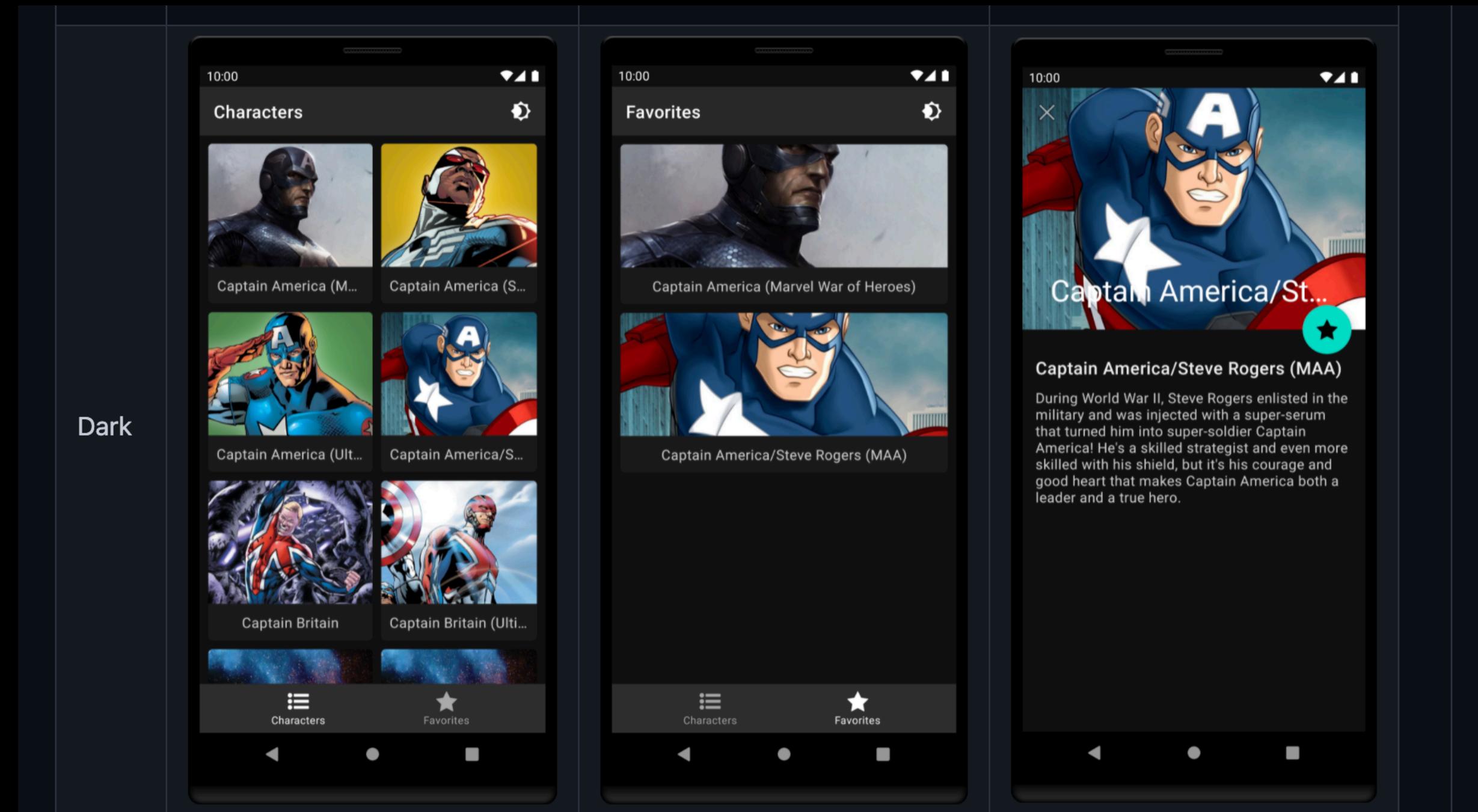
Tech-stack

This project takes advantage of many popular libraries, plugins and tools of the Android ecosystem. Most of the libraries are in the stable version, unless there is a good reason to use non-stable dependency.

Dependencies

- Jetpack:
 - [Android KTX](#) - provide concise, idiomatic Kotlin to Jetpack and Android platform APIs.
 - [AndroidX](#) - major improvement to the original Android [Support Library](#), which is no longer maintained.
 - [Benchmark](#) - handles warmup, measures your code performance, and outputs benchmarking results to the Android Studio console.
 - [Data Binding](#) - allows you to bind UI components in your layouts to data sources in your app using a declarative format rather than programmatically.
 - [Lifecycle](#) - perform actions in response to a change in the lifecycle status of another component, such as activities and fragments.
 - [LiveData](#) - lifecycle-aware, meaning it respects the lifecycle of other app components, such as activities, fragments, or services.
 - [Navigation](#) - helps you implement navigation, from simple button clicks to more complex patterns, such as app bars and the navigation drawer.
 - [Paging](#) - helps you load and display small chunks of data at a time. Loading partial data on demand reduces usage of network bandwidth and system resources.
 - [Room](#) - persistence library provides an abstraction layer over SQLite to allow for more robust database access while harnessing the full power of SQLite.
 - [ViewModel](#) - designed to store and manage UI-related data in a lifecycle conscious way. The ViewModel class allows data to survive configuration changes such as screen rotations.

Some code for you



Challenge for you

- <https://github.com/public-apis/public-apis>
- Clone repo
- Update the code to be able to consume the API of your choice
- I chose: <https://breakingbadapi.com/>
- I chose: <https://unsplash.com/developers>

Official resources

Course structure

4 units with a total of 13 lessons across 13 weeks

Unit 1 (3 weeks)
Get Started with Kotlin
Basics, Functions, Classes & Objects, Extensions

Unit 2 (3 weeks)
Introduction to Android
First App, Layouts, Navigation

Unit 3 (6 weeks)
Android App Architecture
App Architecture, Data Persistence, Display Lists, Connect to Internet, Background Work

Unit 4 (1 week)
App Design
App UI Design

Google Developers Training | Android Development with Kotlin | This work is licensed under the [Apache 2 license](#). | 13

<https://developer.android.com/kotlin?hl=es-419>

Official resources

Lecture slides for instructors

- [Introduction](#) // Android Development with Kotlin
- [Lesson 1](#) // Kotlin Basics
- [Lesson 2](#) // Functions
- [Lesson 3](#) // Classes and Objects
- [Lesson 4](#) // Build your first Android app
- [Lesson 5](#) // Layouts
- [Lesson 6](#) // App Navigation
- [Lesson 7](#) // Activity and Fragment Lifecycle
- [Lesson 8](#) // App Architecture (UI Layer)
- [Lesson 9](#) // App Architecture (Persistence Layer)
- [Lesson 10](#) // Advanced RecyclerView use cases
- [Lesson 11](#) // Connect to the Internet
- [Lesson 12](#) // Repository pattern and WorkManager
- [Lesson 13](#) // App UI Design

What's included?

- Lecture slides with speaker notes
- Hands-on codelabs for students (presented in pathways)
- Digital badges for students
- Solution code on GitHub

Prerequisites

Familiar with object-oriented programming language, how to use an IDE, and how to use GitHub. No prior experience with Kotlin or Android is necessary. It is recommended that instructors complete the course content before teaching it.

Learning pathways for students

[Explore all pathways](#)

Official resources

Pathway

Practice what you've learned by completing the pathway:

[Lesson 13: App UI Design](#)



Google Developers Training | Android Development with Kotlin | This work is licensed under the [Apache 2 license](#). | 59

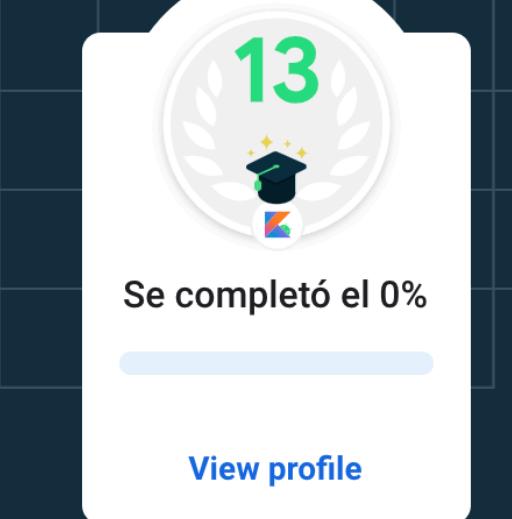
<https://developer.android.com/courses/pathways/android-development-with-kotlin-13>

Official resources

Lesson 13: App UI design 

Learn UI best practices and how to make your app follow the Material Design guidelines.

4 actividades • 1 cuestionarios



View profile

- 1 **Material Components Basics**  Codelab
- 2 **Material structure and layout**  Codelab
- 3 **Material theming with color, elevation and type**  Codelab
- 4 **Android styling: common theme attributes**  Codelab

Quiz

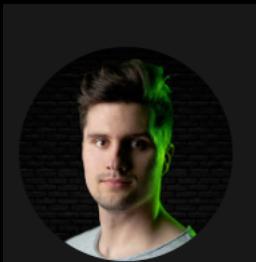
<https://developer.android.com/courses/pathways/android-development-with-kotlin-13>

Recommended resources

The screenshot shows a course page on the Udemy website. At the top, there's a search bar and navigation links for 'Udemy Business' and 'Teach on Udemy'. Below the header, the course path is shown as 'Development > Mobile Development > Android Development'. The main title of the course is 'Master Desarrollo Android con Java y Kotlin [Abril 2020]'. A brief description follows: 'Desarrolla apps Android AVANZADAS Java/Kotlin desde cero: Android Studio, Jetpack, Firebase, Maps, AdMob, Room, APIs,...'. The course has a rating of 4.1 stars from 1,434 ratings and 7,409 students. It was created by Alejandro Lora, Miguel Campos Rivera, and The Good Code Academy, last updated on 4/2020. Language options include Spanish and Spanish [Auto]. Action buttons for 'Wishlist' (with a heart icon), 'Share' (with a share icon), and 'Gift this course' are visible. To the right, there's a large promotional image featuring a colorful Android robot, various development tools like Git, Java, and Firebase, and a play button icon. The price of the course is listed as \$89.99. Below the price are two buttons: 'Add to cart' (in purple) and 'Buy now' (in white). A small note at the bottom states '30-Day Money-Back Guarantee'.

<https://www.udemy.com/course/programacion-android-avanzado-de-principiante-a-experto/>

Recommended resources



Philipp Lackner
36,100 suscriptores

<https://www.youtube.com/c/PhilippLackner/featured>



Coding in Flow •
183,000 suscriptores

<https://www.youtube.com/c/CodinginFlow/featured>



Sociedad Androide
12,300 suscriptores

<https://www.youtube.com/c/SociedadAndroide/featured>

“Make a 1million dollars”

-Nobody

Q&A

<https://github.com/normanaspX/ieee-conference>

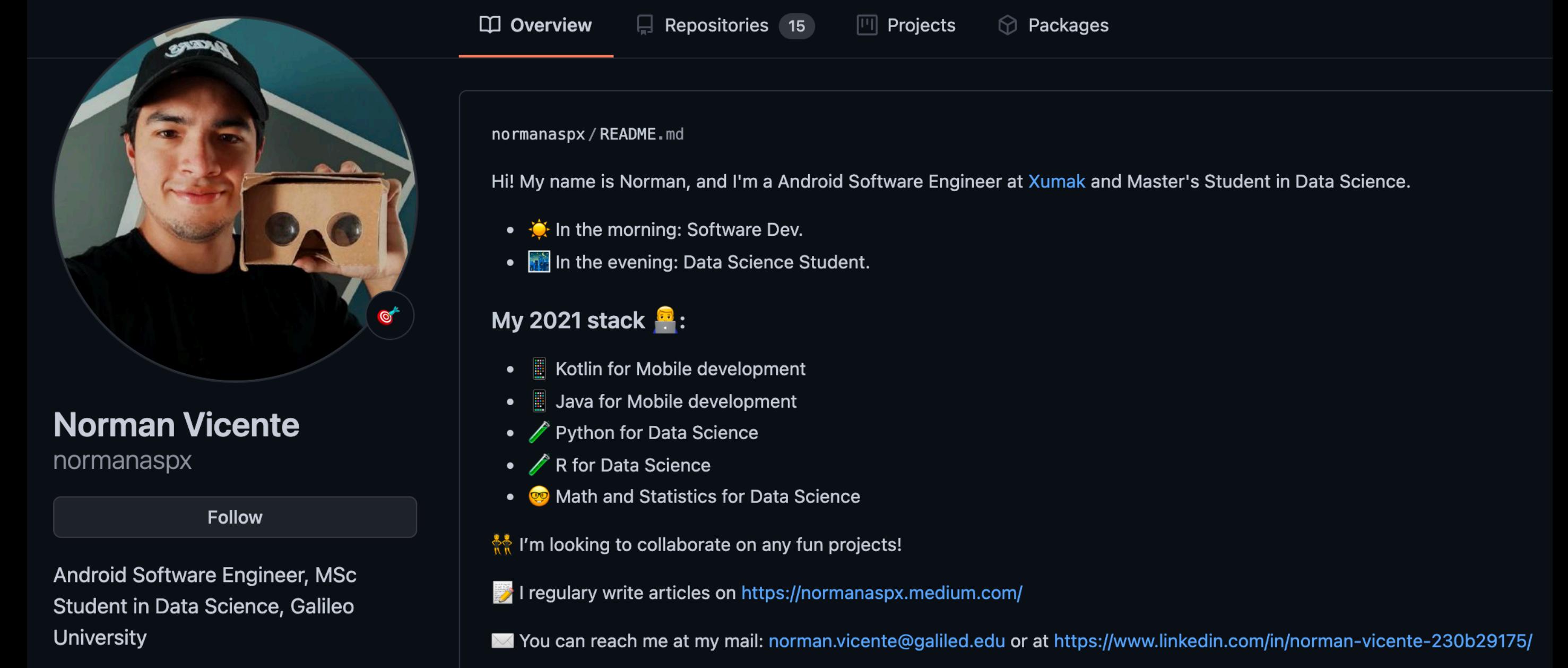
Reach me

LinkedIn: @normanaspx

GitHub: @normanpasx

Facebook: @normanaspx

Medium: @normanaspx



A screenshot of a GitHub profile page for the user 'normanaspx'. The profile picture shows a man wearing a cap and holding a Google Cardboard VR viewer. The profile name is 'Norman Vicente' and the handle is 'normanaspx'. Below the name, it says 'Android Software Engineer, MSc Student in Data Science, Galileo University'. There is a 'Follow' button. At the top of the page, there are tabs for 'Overview', 'Repositories 15', 'Projects', and 'Packages'. The 'Overview' tab is selected. The bio section starts with 'normanaspx / README.md' and says 'Hi! My name is Norman, and I'm a Android Software Engineer at Xumak and Master's Student in Data Science.' It lists two bullet points: 'In the morning: Software Dev.' and 'In the evening: Data Science Student.'. The 'My 2021 stack' section lists five items: 'Kotlin for Mobile development', 'Java for Mobile development', 'Python for Data Science', 'R for Data Science', and 'Math and Statistics for Data Science'. There are also three footer notes: 'I'm looking to collaborate on any fun projects!', 'I regularly write articles on <https://normanaspx.medium.com/>', and 'You can reach me at my mail: norman.vicente@galiled.edu or at <https://www.linkedin.com/in/norman-vicente-230b29175/>'.

<https://normanaspx.github.io/norman/>