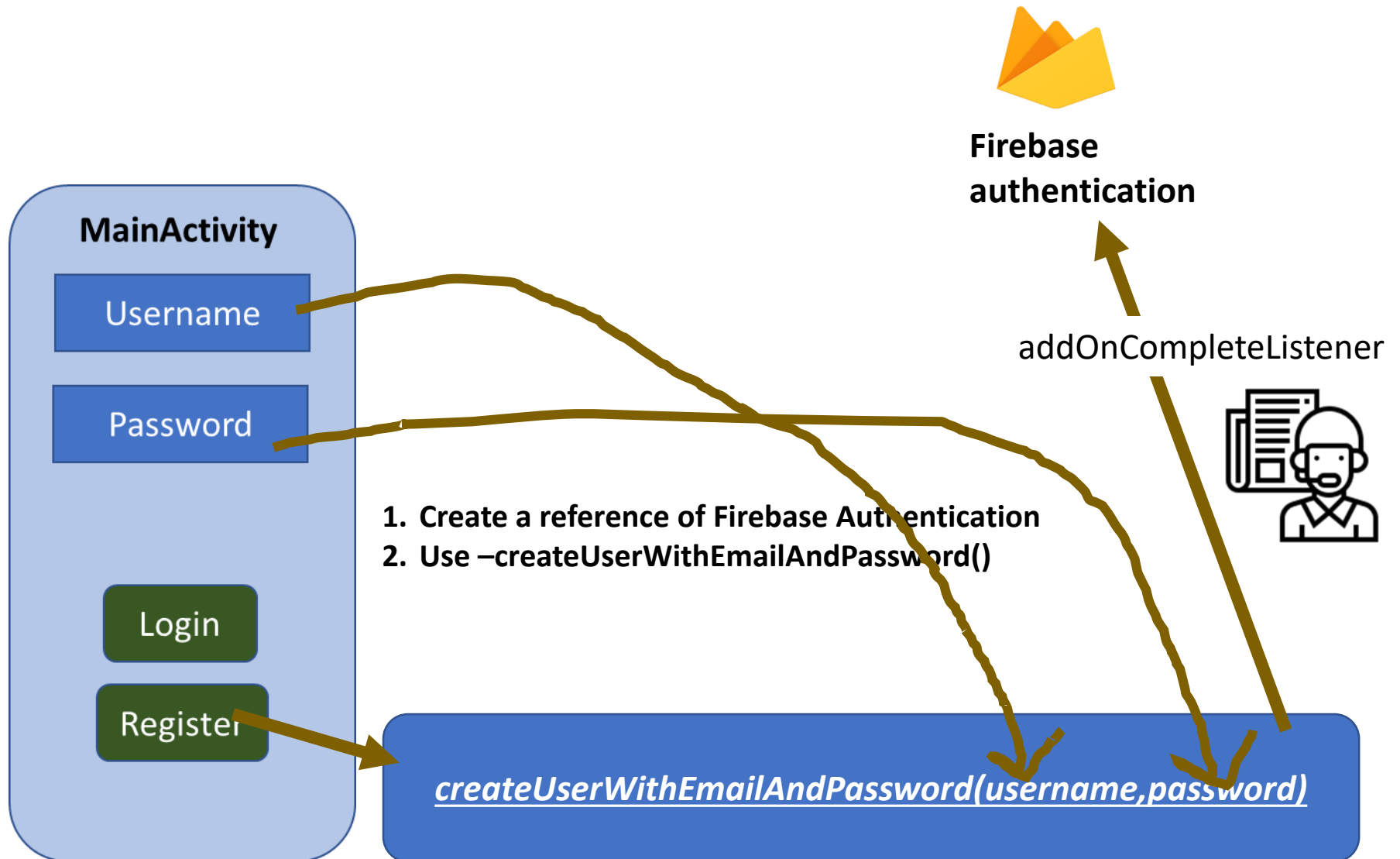


Summer 2, 2019 - CS 4520/CS5520 – Mobile Application Development

Pratheep Kumar Paranthaman, Ph.D.,





User Authentication

User Authentication

- Authenticate only the registered users to enter the App

signInWithEmailAndPassword(username,password)

- **Set EventListeners to the task**
 - **onCompleteListener**

Storing and retrieving files

Storing and retrieving files from Firebase

- In this part we'll practice of store files(Images) to the **STORAGE** unit in Firebase and extract it.

~~Authentication – manages user registration and authentication~~

~~Realtime database – Cloud hosted NoSQL Database~~

Storage – storage management for various files(images, documents)

Hosting – Global web hosting

MLKit – SDKs for ML tasks

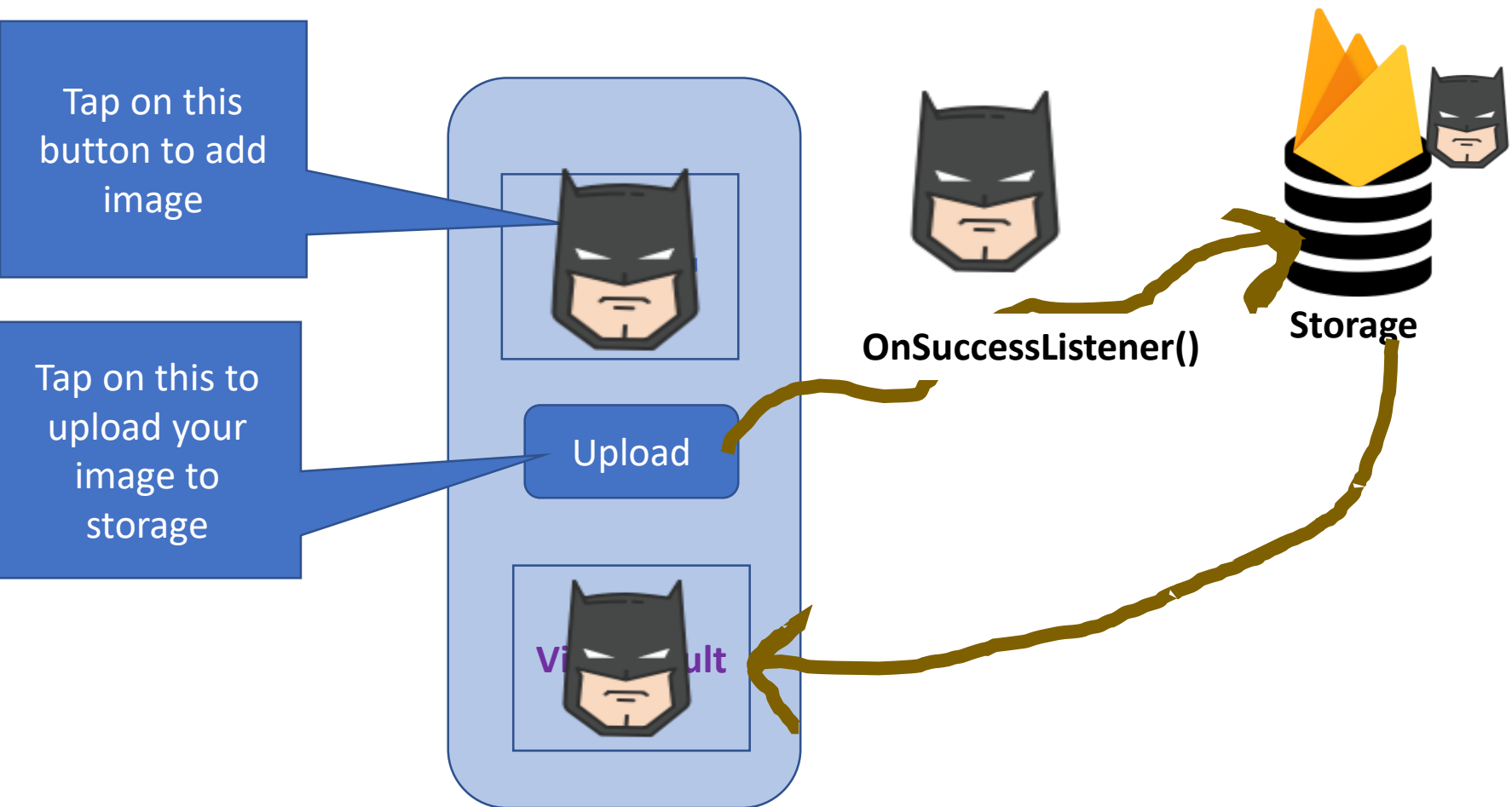
Firebase Storage

- Efficient for storing documents, images, and other multimedia files
- All the contents can be synchronized across various users.
- The security rules can be set to manage authentication and data validation

Exercise - 8



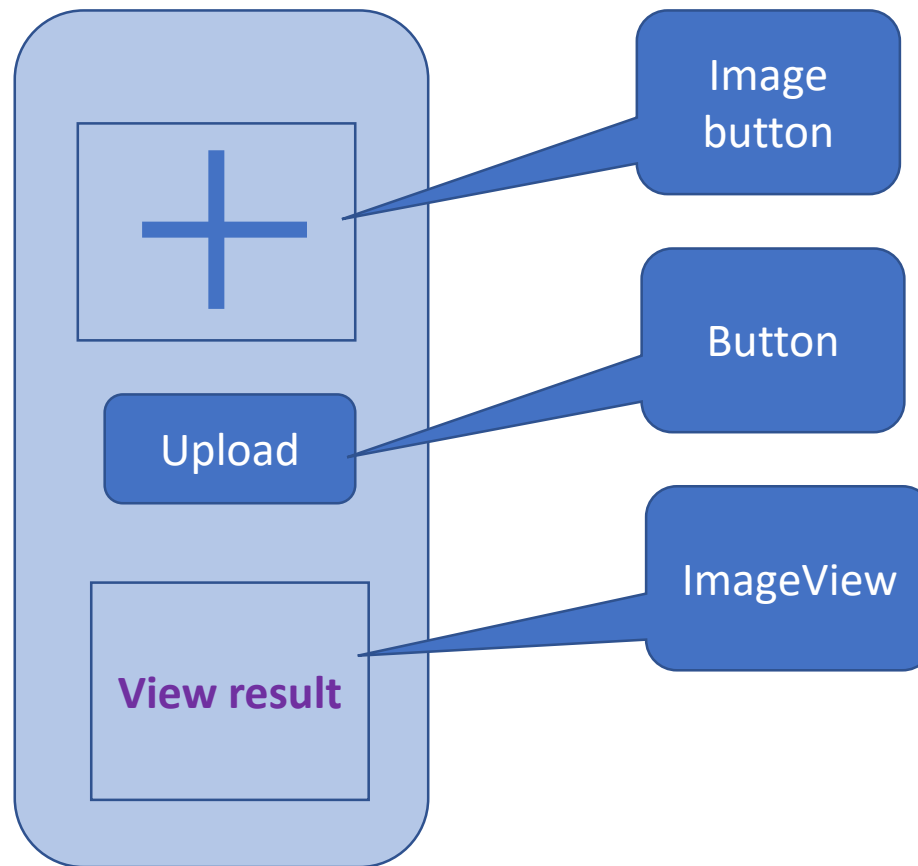
Exercise 8 – File Storage and Retrieval



Configuration

Implementation 'com.google.firebase:firebase-storage:18.1.0'

Layout creation

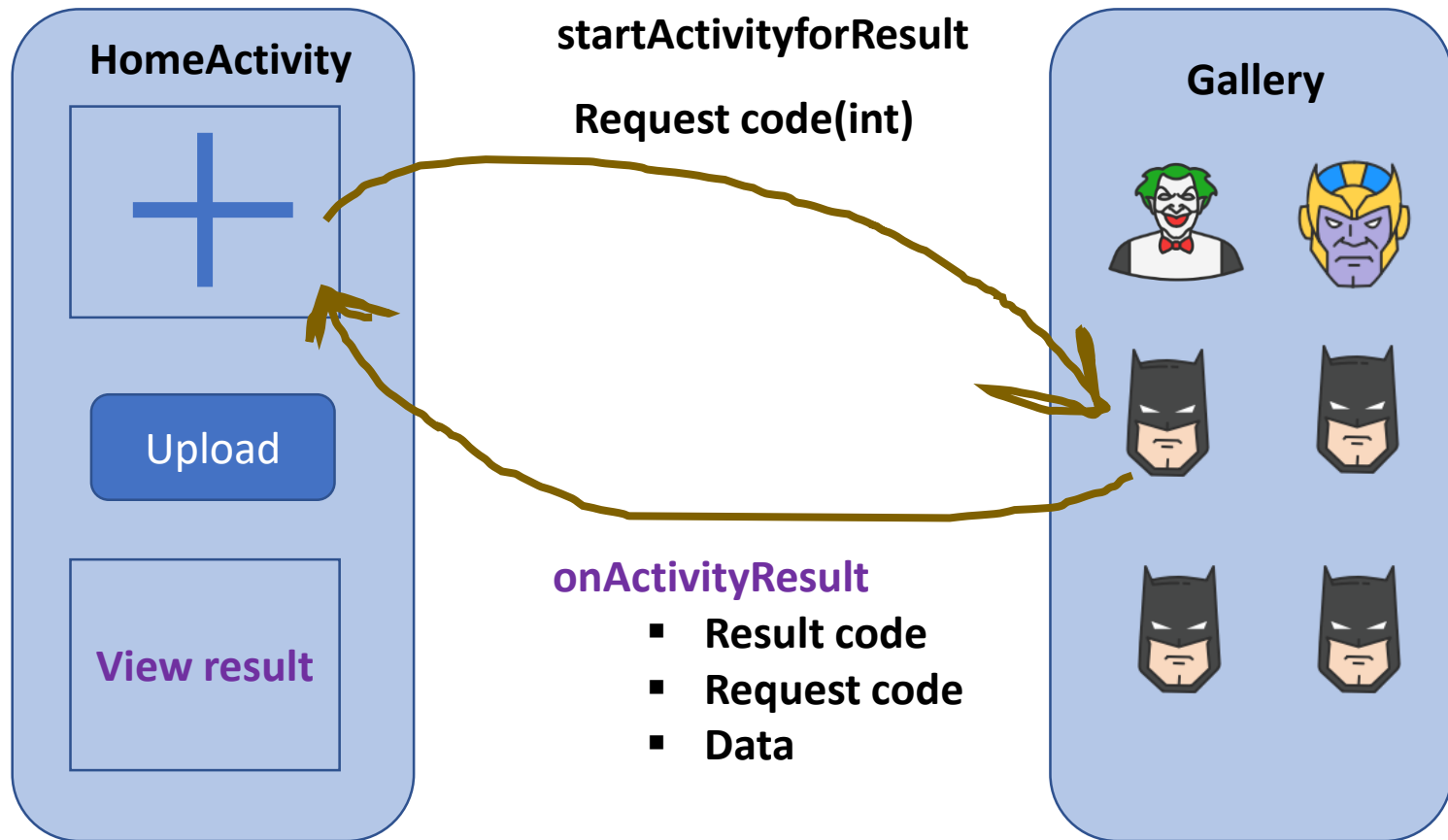


Storage and Retrieval

- ~~Layout creation~~
- **Upload the Image – Storage process**
 - Use Implicit intents to open Gallery
 - Extract the image from gallery and give it back to your calling (Activity)
 - **Uploading to Firebase**
 - **Include the dependencies**
 - Get the reference StorageDatabase -> StorageReference
 - Use putFile() -> pass in the ImageURI

Implicit Intents

Implicit Intent(ACTION_GET_CONTENT)



Implicit Intents to get Result from Activity

- Declare Implicit Intent -> ACTION_GET_CONTENT
- Set intent Type("image/*")
- StartActivityForResult(ImplicitIntent, Request Code)
- When user selects an Image
 - The result will be returned in onActivityResult()
 - Returns the **ImageURI (data)** -> **URI??????**
 - Update the RESULT Image on the Image button

Upload File

Steps for Uploading

- **Get the reference of the service(FirebaseStorage)**
- Create a reference path for your image -> Based on image URI
- PutFile in Storage
- **add OnSuccessListener**

Get Reference of Storage

```
FirebaseStorage storageReference = FirebaseStorage.getInstance().getReference();
```

Upload the Image

- Create a reference path ->

The screenshot illustrates the Google Cloud Storage (GCS) interface. The top navigation bar shows 'Storage' with tabs for 'Files', 'Rules', and 'Usage'. The main content area displays the path 'gs://testprojectfb-f1fca.appspot.com'. A red box highlights the 'MyImages/' folder, which is labeled 'Child' in a blue callout. Below this, the 'Image files' are listed in a table, with a red box highlighting the first four rows, labeled 'Image files' in a blue callout. The table columns are 'Name', 'Size', 'Type', and 'Last modified'.

Name	Size	Type	Last modified
127	904.33 KB	image/jpeg	Jul 21, 2019
128	90.79 KB	image/jpeg	Jul 21, 2019
6503209	16.2 KB	image/png	Jul 22, 2019
image:62	16.2 KB	image/png	Jul 22, 2019

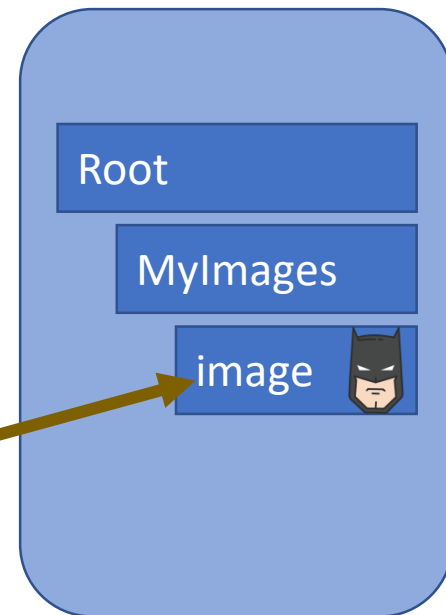
Create a reference path

```
final StorageReference filepath= storageReference.child("MyImages")  
.child(imageURI.getLastPathSegment());
```

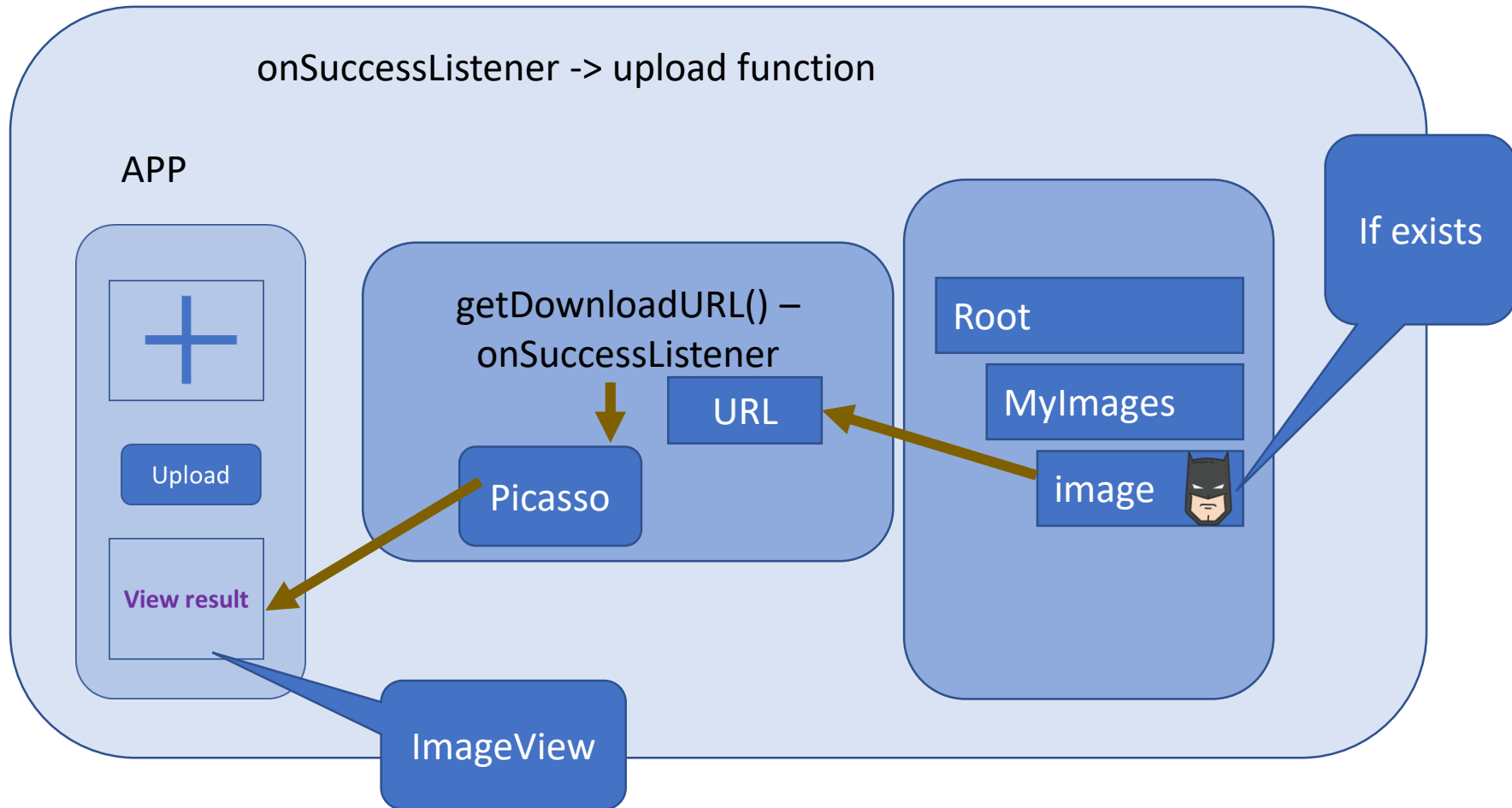
Image
name

addOnSuccessListener

filePath.putFile(ImageURI)



Extract the Image as soon as it gets uploaded

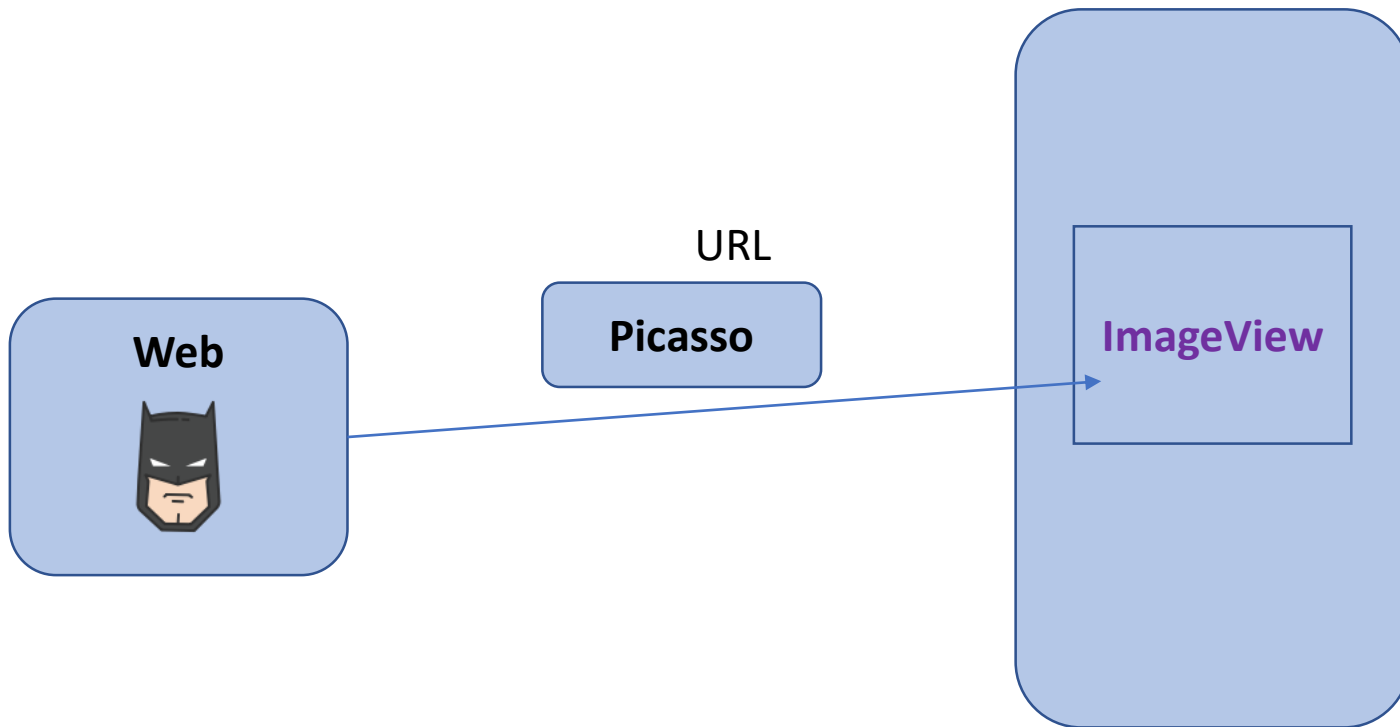


How to get your Image URL from Firebase?

- If your file exists in the firebase store path then call
 - `getDownloadURL()` on your `FilePath`
 - `SetOnSuccessListener` -> to extract the URL

Picasso

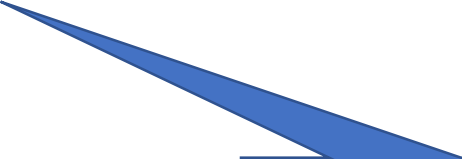
- Library for image downloading and caching
- Allows you to download image from a URL and load it onto your *ImageView*



Picasso Library

<https://github.com/square/picasso>

implementation 'com.squareup.picasso:picasso:2.71828'



Add this in
Build.gradle(module)

Firebase documentation

- https://firebase.google.com/docs/android/setup#available_libraries