ISIMA
INSTITUT D'INFORMATIQUE
Université Clermont Auvergne

Android - Introduction

ISIMA - 22/01/2019

Thomas Coulange - Olivier Goutet

openium
créateur d'applications

Summary

- Origins and nowadays
- Our App example
- App architecture
- Android development
  - Layout
  - Permissions
  - Content Providers
  - Lists
  - Images
  - App navigation
  - Storage
- Misc
- Useful libraries

Origin and nowadays

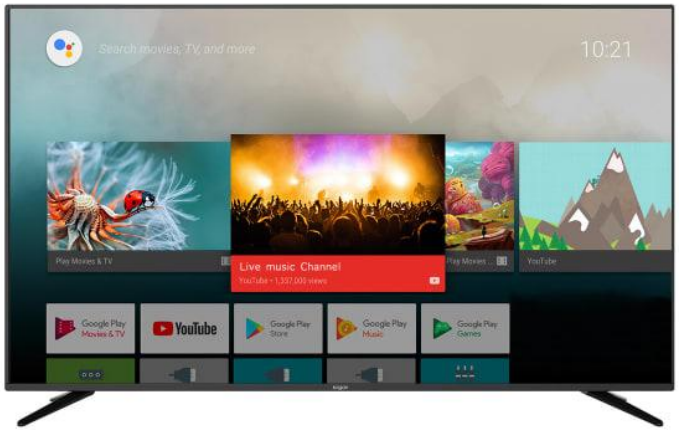
Android origin

- Mobile operating system
  - Based on Linux Kernel
  - With little of OpenSource
- Developed by Android Inc. since 2003
- Bought by Google in 2005
- Developed by Google since then
- Developed by the Open Handset Alliance since 2007
  - Texas Instruments, Qualcom, Broadcam, Intel, Nvidia...
  - Google, Nuance, NXP Software...
  - Samsung, LG, HTC, Sony, Motorola, Huawei, ZTE…
  - Vodaphone, TMobile, China mobile, Bouygues Telecom…

First Android phone 2008

~ Latest Android phone

Not only phones



UI evolution

Versions history

Customisation 定制化服务

- Most of manufacturers customize the stock Android
  - Samsung with One UI
  - Xiaomi MIUI
  - Huawei Emotion UI
  - …
- Some don’t (or not much)
  - OnePlus
  - Nokia
  - Lenovo (Motorola)
- Android Strength
  - Users can choose the UI there prefer

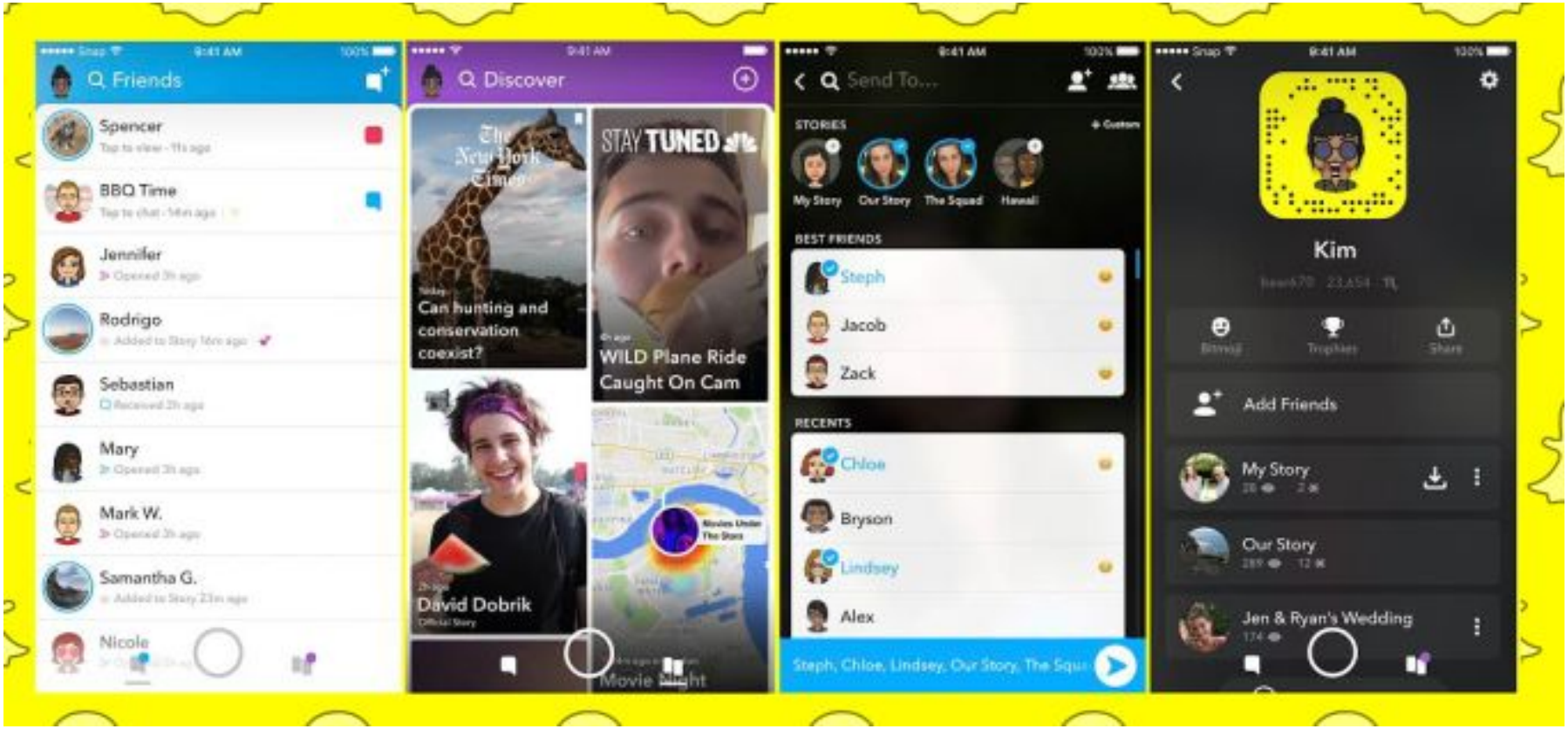
Customisation problems

- Not only UI modifications
- Some manufacturer goes too far
  - Not all phones works the way described in the Android documentation
- Widgets with strange behaviour
- Aggressive battery optimization
  - Randomly kill apps
  - Major problems for developers
- Majors culprits 犯过错者，被告人
  - Xiaomi
  - Huawei
  - OnePlus
  - Nokia

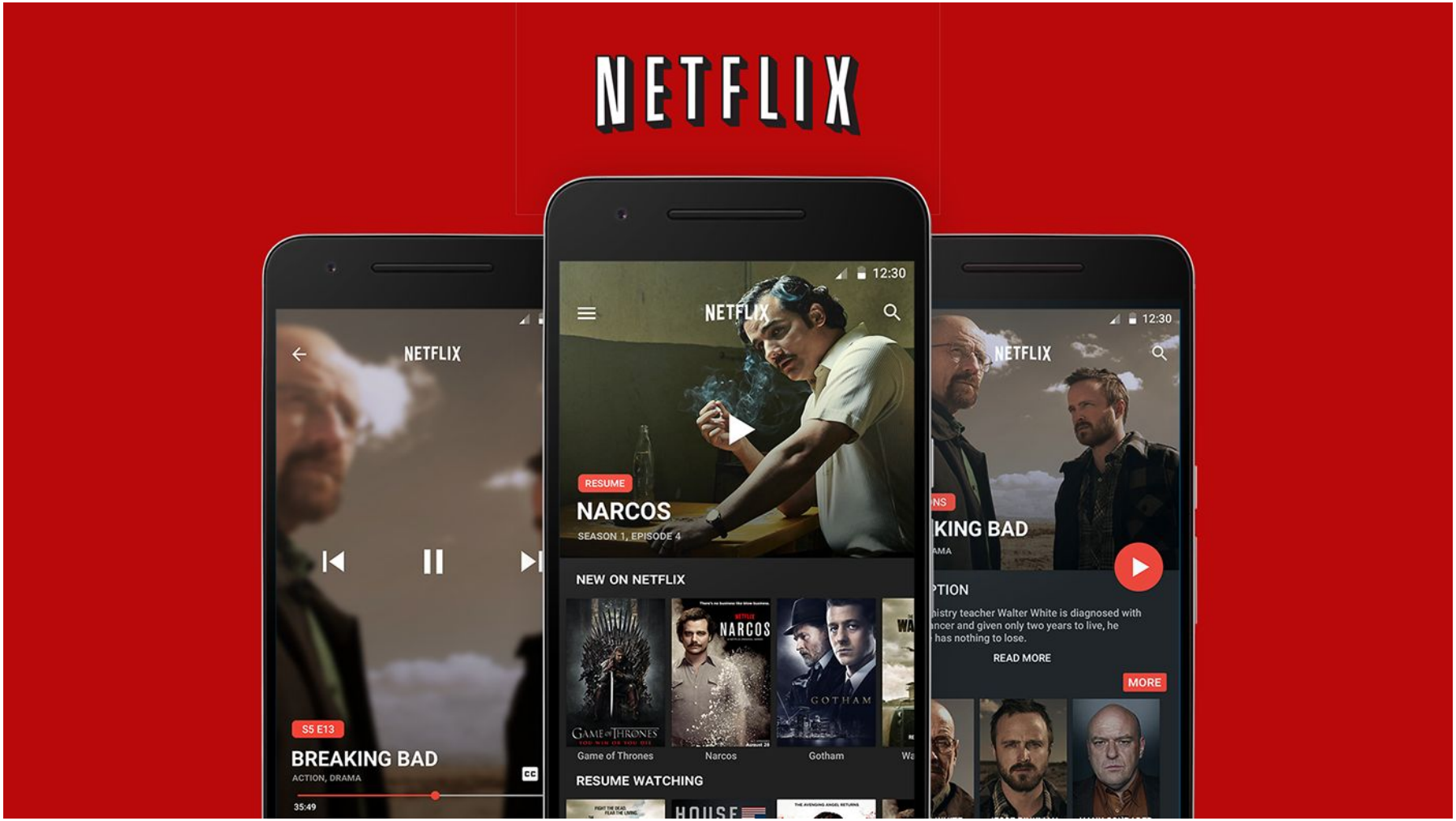
Our App example

Apps everybody wants to develop

Apps everybody wants to develop



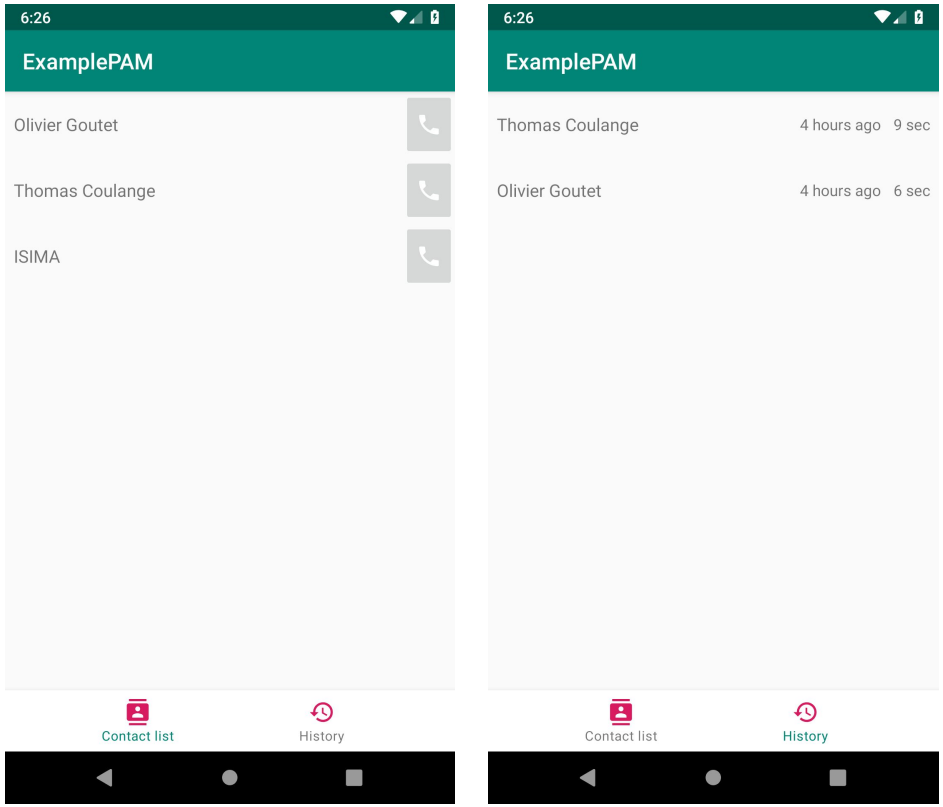
Apps everybody wants to develop



Apps everybody wants to develop

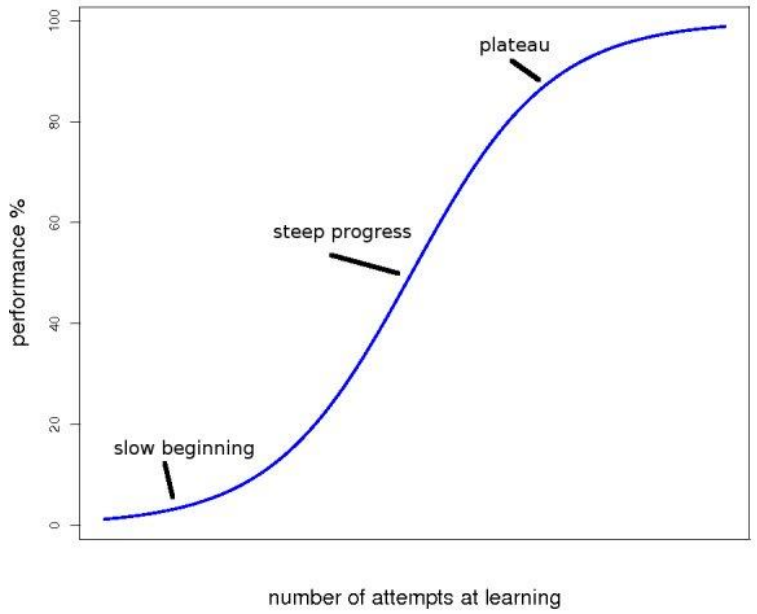
Simple call app

Source available at
https://github.com/openium/examplepam



Simple call app

- First use of many Android components
  - Activity/Fragment
  - Permissions
  - Layouts
  - Widgets (Buttons, TextView, ImageView…)
  - Content providers
- Good start for Android development



App architecture

Architecture of Android Apps

- Source code - src folder
  - Activity - Screen/Controller
    - Fragment - Part of Screen/Controller
    - View
  - Context - Provide access to resources of your app, and system APIs
  - Your code!

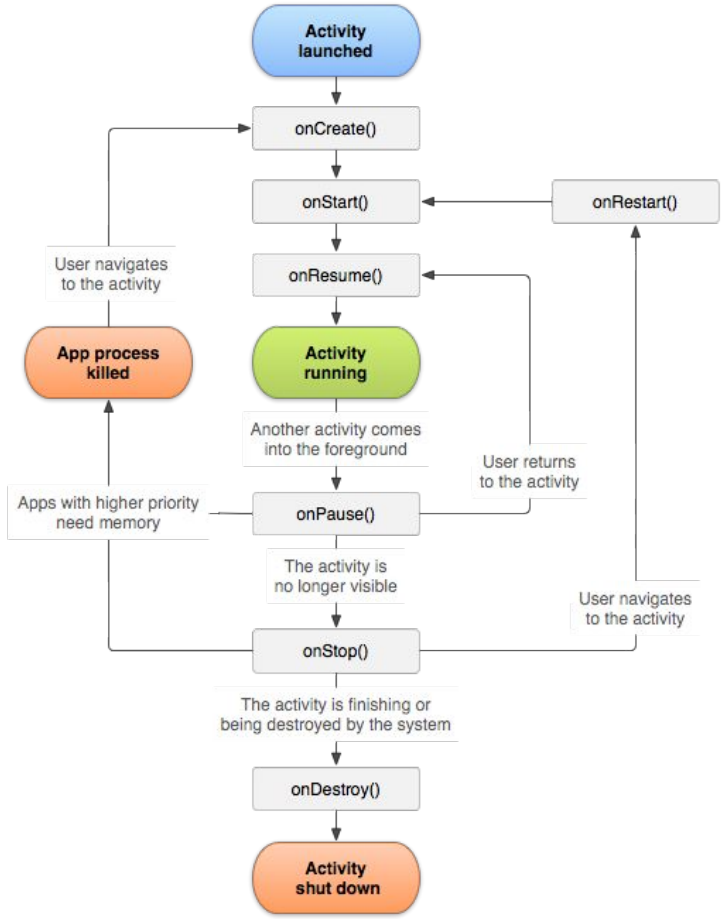
- Resources - res folder
  - Colors
  - Drawable
  - Strings
  - Style
- Build & Manifest
  - AndroidManifest.xml
  - build.gradle

Architecture of Android Apps

Demo

Activity

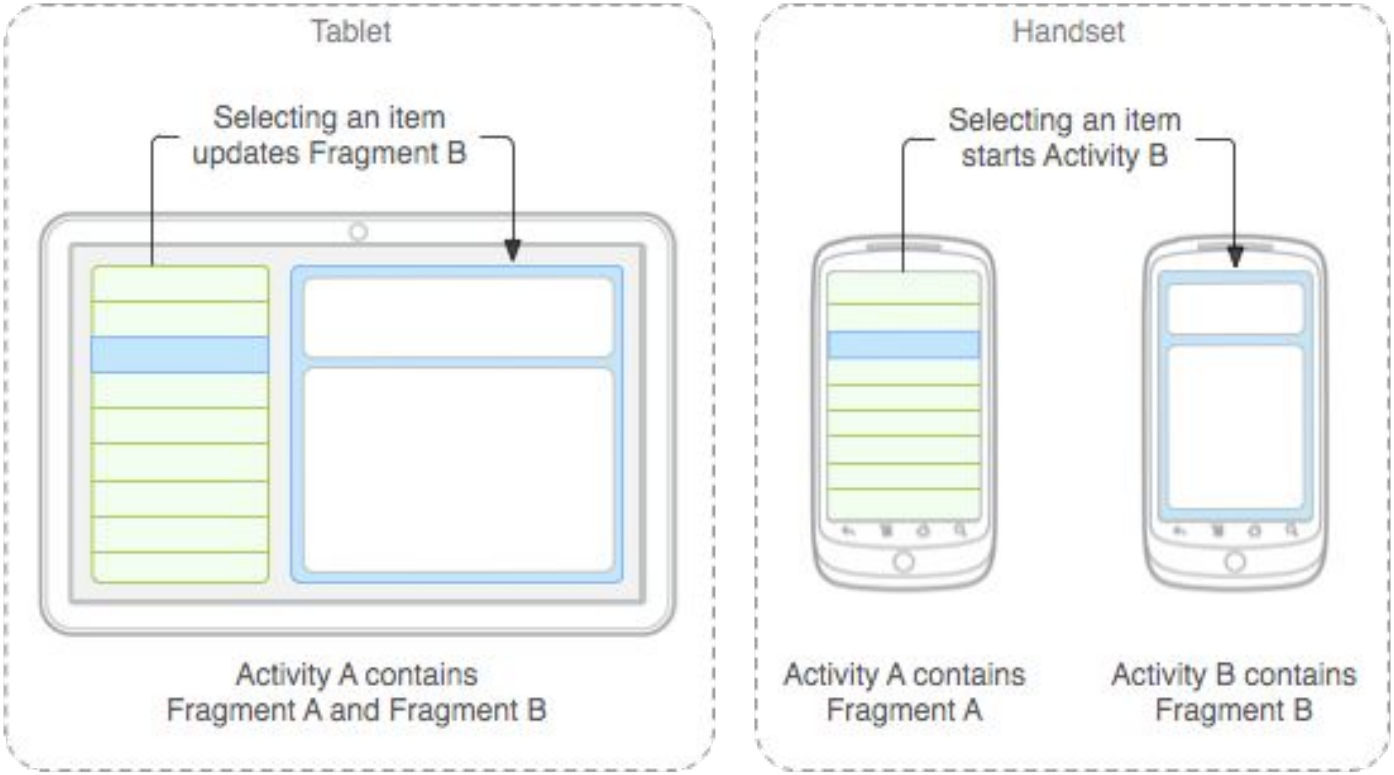
- Represent the screen displayed to the user
- Only one activity can be present at any time
- Has a specific lifecycle to know what the user is doing
- On rotation the activity is destroyed then recreated



Fragment

- Is contained by an activity
- There can be multiple fragments in one activity
- Has a different lifecycle than the activity
- Can be considered as a “complex” View

Fragment


Tablet
Selecting an item
updates Fragment B
Activity A contains
Fragment A and Fragment B
Handset
Selecting an item
starts Activity B
Activity A contains
Fragment A
Activity B contains
Fragment B

Resources

- Stored in res folder of project
- Multiple type (strings, colors, images)
- Require a Context to get from your code
- Can use qualifier (device size, lang)
- Most files are declared in XML

Resources

Demo

Layout

Layout

A ViewGroup organizing its child View.

Every activity/fragment will use a layout file to describe its content.

It is stored in the layout folder of your resources, and wrote in XML

Types

- LinearLayout
  - Views are put one after the other, horizontally or vertically
- RelativeLayout
  - Each View is placed in a relative way to another view
- FrameLayout
  - Views are placed in a relative way to the layout
- ConstraintLayout
  - Complex and new layout meant to replace the RelativeLayout with tons of cool feature

Layout

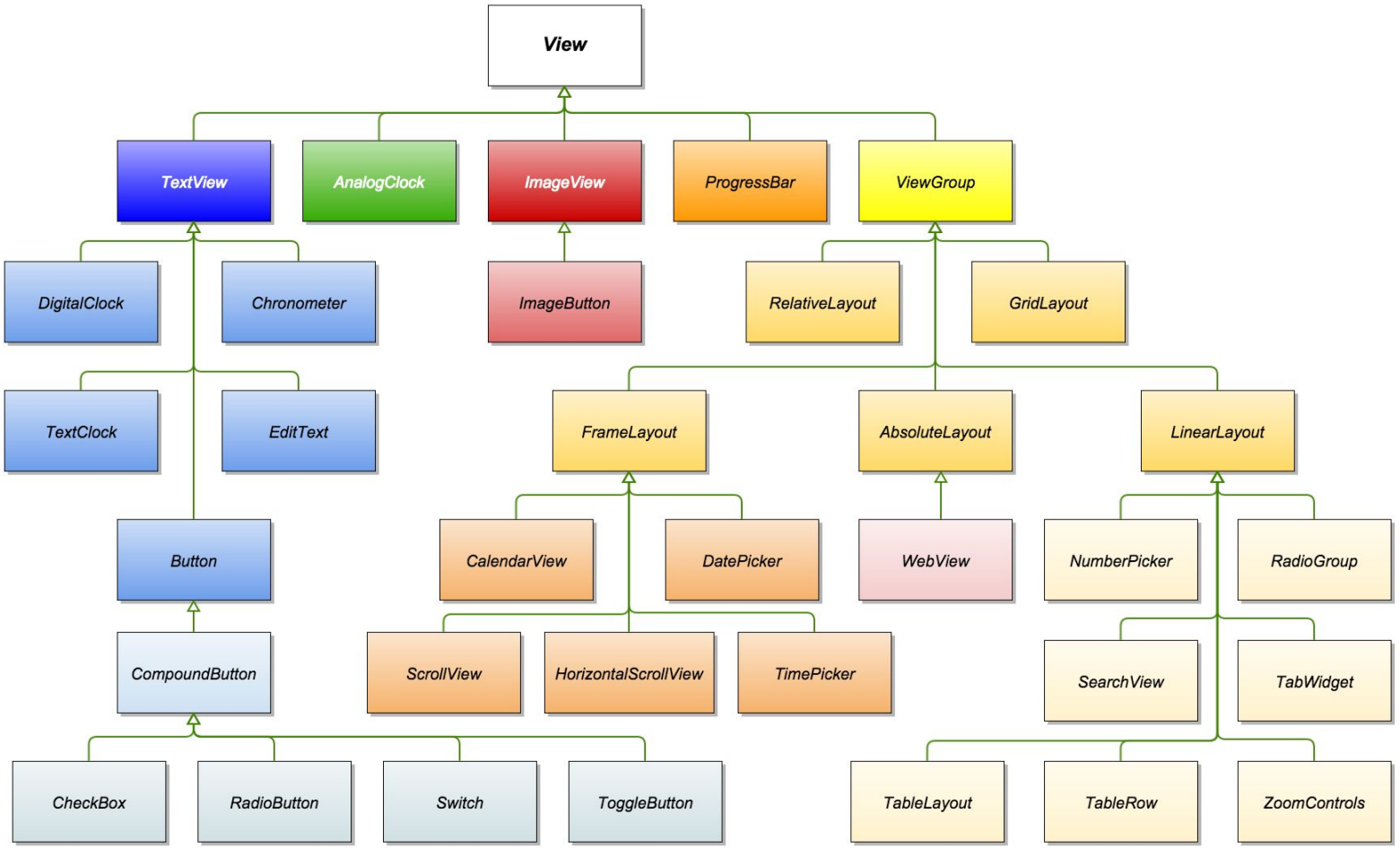
Demo

View

Types

- TextView - display text
- EditText - allows user input to enter text, inherit TextView
- ImageView - display image
- Many more, all extending the View class

View


View
TextView
AnalogClock
ImageView
ProgressBar
ViewGroup
DigitalClock
Chronometer
ImageButton
RelativeLayout
GridLayout
TextClock
EditText
FrameLayout
AbsoluteLayout
LinearLayout
Button
CalendarView
DatePicker
WebView
NumberPicker
RadioGroup
CompoundButton
ScrollView
HorizontalScrollView
TimePicker
SearchView
TabWidget
CheckBox
RadioButton
Switch
ToggleButton
TableLayout
TableRow
ZoomControls

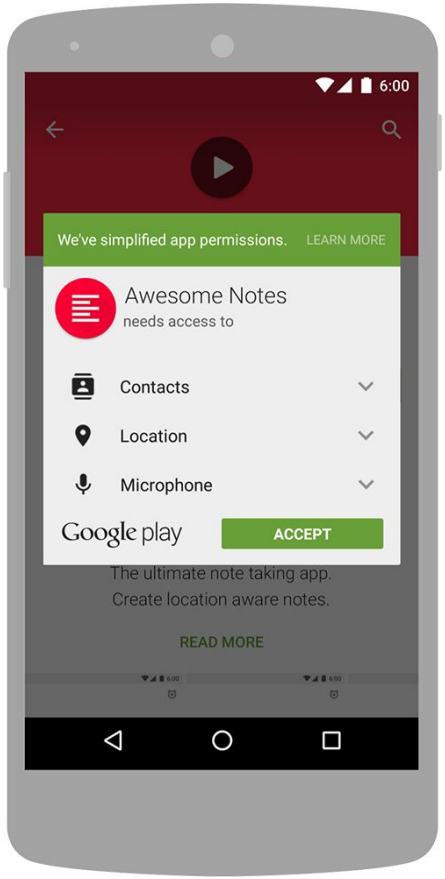
Permissions

Permissions

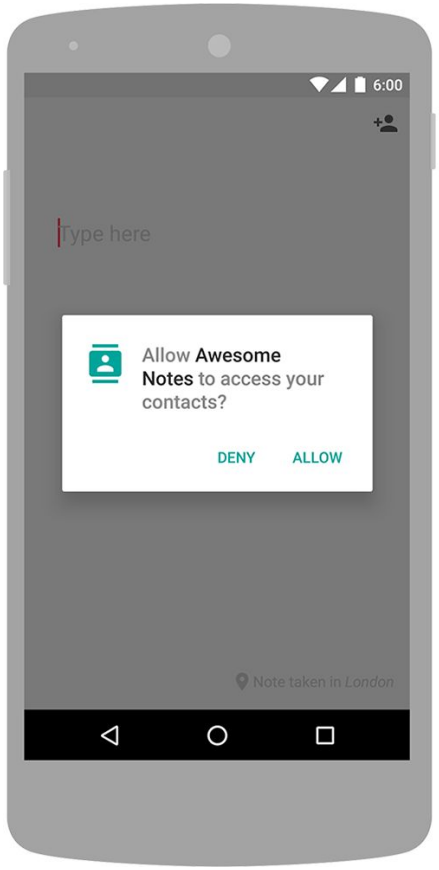
- Android security system is based on permissions
- Explicit declaration of functionality needed by the app
- User validation to accept the permissions
  - Because it’s based on user, it can be a problem...

Permissions

- Since Android 1.0
  - Install-time request permissions



- Since Android 6.0
  - Runtime permissions



Permissions

- Normal permissions
  - Granted at install time
  - INTERNET
  - BLUETOOTH
  - VIBRATE
  - ACCESS_NETWORK_STATE
  - ...
- Dangerous permissions
  - Granted with explicit user agreement
  - READ_CALENDAR / WRITE_CALENDAR
  - READ_CALL_LOG / WRITE_CALL_LOG
  - RECORD_AUDIO
  - CAMERA
  - SEND_SMS / READ_SMS
  - WRITE_EXTERNAL_STORAGE
  - ...

Permissions

Demo

Content providers

What is a Content Provider

- A way of sharing data between apps
- Often linked to an SQLite storage
- The old way to store and access data in your own app

Accessing data from others app

- Usage of the content resolver
  - A service that connect to the content providers of all the app installed on the phone
- Use SQL queries to get the content on an URI (**Uniform Resource Identifier**)
- Return a Cursor
  - An object that provide random read and write access on the query
- In our app, we request the user's contacts from a system Content Provider

Content Provider

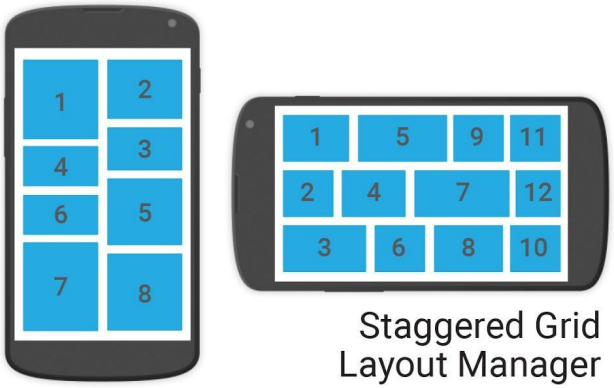
Demo

Lists

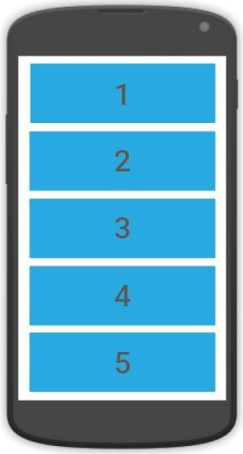
RecyclerView

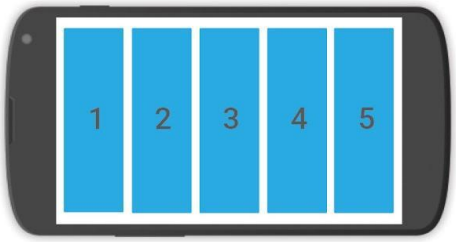
- Called Recycler because it reuse cells
- Can display elements as horizontal/vertical list, or grid, called LayoutManager
- Except for choosing the LayoutManager, your code will only use the Adapter Class

Layout Manager

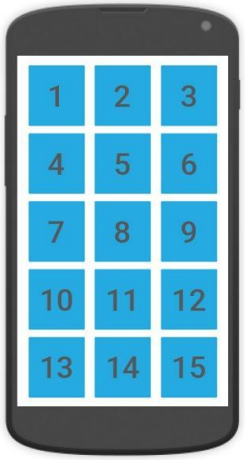


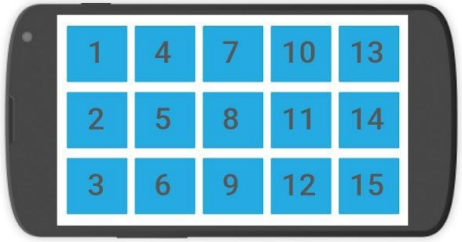
Staggered Grid Layout Manager





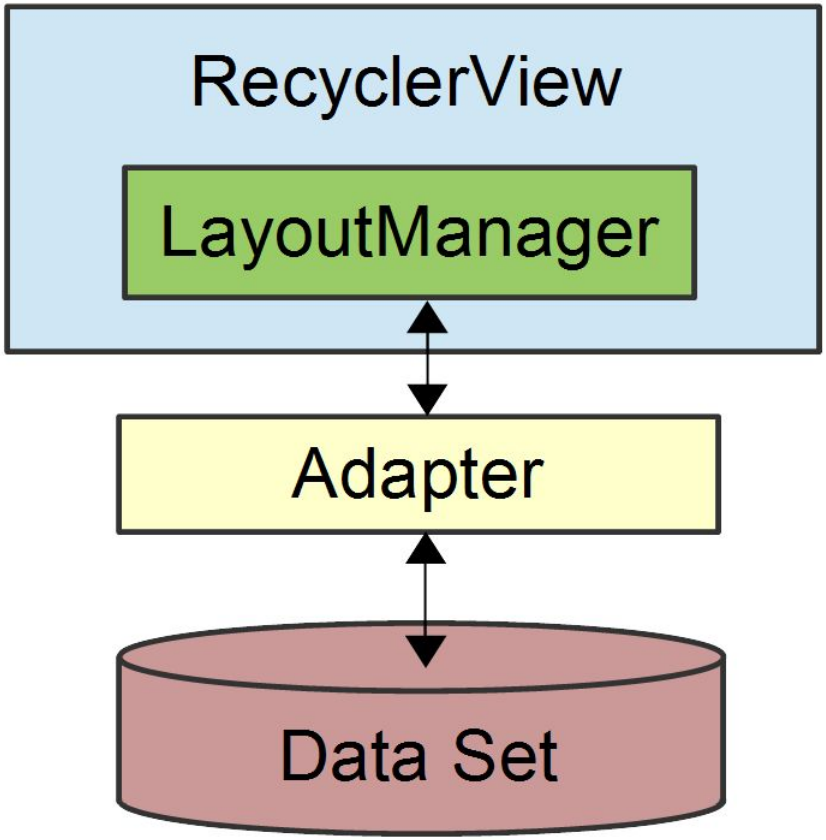
Linear Layout Manager





Grid Layout Manager

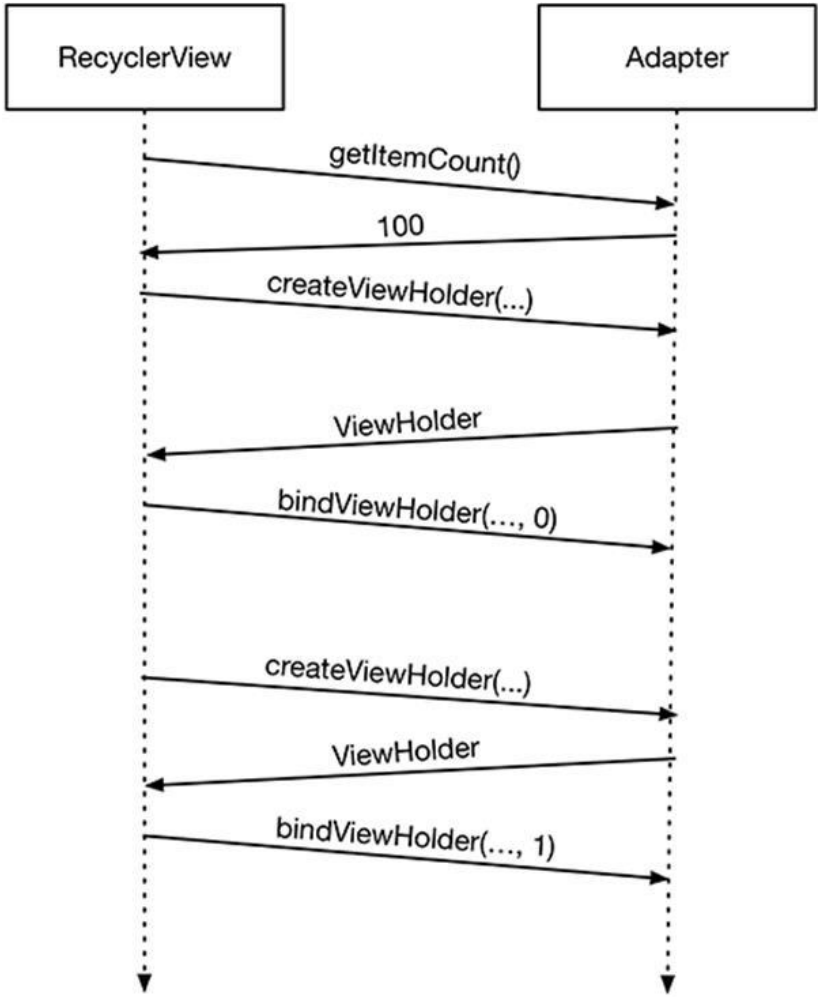
RecyclerView and Adapter



Adapters

- The core of displaying list in an app
- The class Adapter must be extended and passed to the Recyclerview
- The RecyclerView will then call 3 methods on your adapter
- **getItemCount()** which provides the number of cells to display in your list
- **createViewHolder()** which creates as many View as visible cells in your screen
- **bindViewHolder(position)** which bind the view with the data for the position

RecyclerView and Adapter



Apprize.info

Images

Images


08:00
Entreprendre
Compteurs Linky : où en est-on du déploiement en Corrèze ?
Le 19 janvier 2019 à 08H00
Depuis décembre 2015, les compteurs Linky sont déployés dans toute la Corrèze. Fin 2020, près de 166.000 boîtiers verts seront installés dans les foyers et établissements publics. Une question se pose aujourd'hui : où en est-on à l'heure actuelle ?
D'Ussel à Brive, en passant par Tulle, Égletons, Lubersac ou encore Eygurande. Le déploiement des compteurs Linky fait son bonhomme de chemin. Près de 92.000 ont déjà été installés et ce n'est pas (encore) terminé.
92.000
Au 10 janvier 2019, environ 92.000 compteurs Linky

Images

- DIY (Do It Yourself)
  - HttpUrlConnection
  - InputStream
  - BitmapFactory
  - Bitmap → ImageView
- Can be problematic
  - Memory Management
  - ListView/RecyclerView…
- Libraries
  - **Picasso**
  - Glide
  - Fresco
  - Volley
  - ...

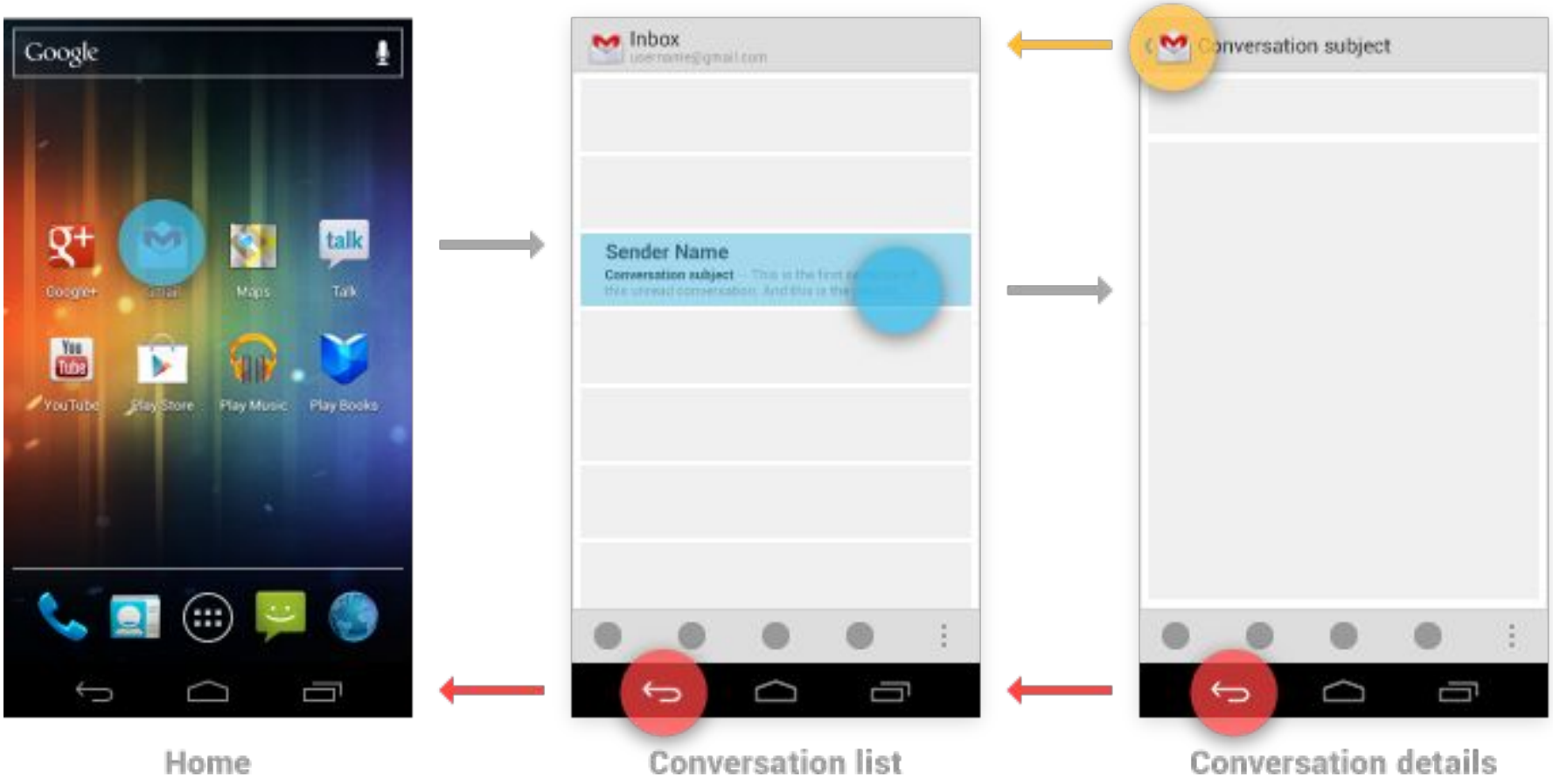
Picasso

```
Picasso.get().load(url).into(imageView1);
```

```
Picasso.get()
    .load(url)
    .placeholder(R.drawable.user_placeholder)
    .error(R.drawable.user_placeholder_error)
    .into(imageView);
```

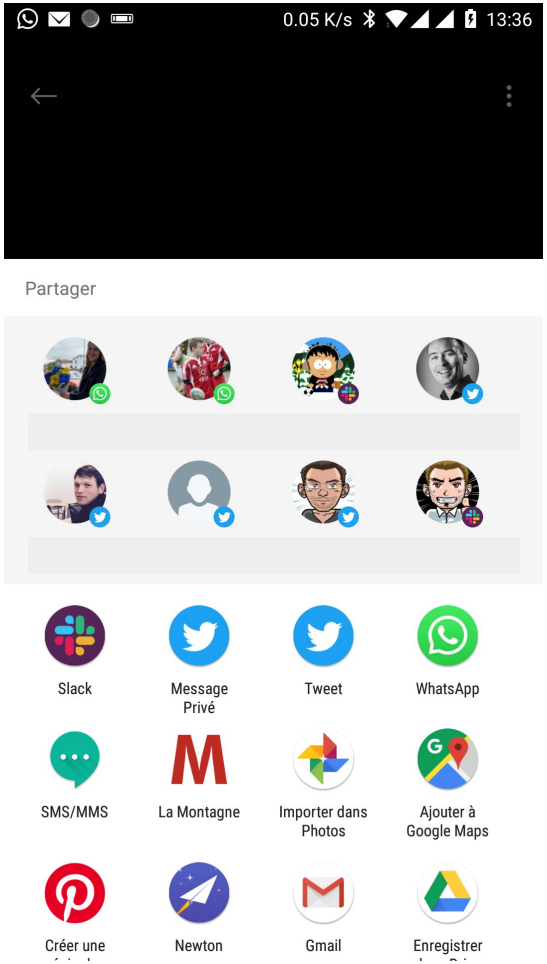
App navigation

Activity


Google
Google+
Maps
Talk
YouTube
Play Store
Play Music
Play Books
Inbox
Sender Name
Conversation subject
Conversation subject
Home
Conversation list
Conversation details

Intent

- Allows communication with the system
- Example of use
  - Start your trip on Google Maps
  - Phone call your buddy
  - Share data on Twitter
  - Send an SMS
  - Pick a picture in your gallery
  - …
- Example with Picture Sharing



Intents

- In the official documentation :
  An intent is an abstract description of an operation to be performed.

- Message to an
  - Activity (our application or another)
  - Service
  - BroadcastReceiver

- It's a fundamental tool on Android. It allows different applications of different developers to communicate and work together in a loose coupling maner.

Storage

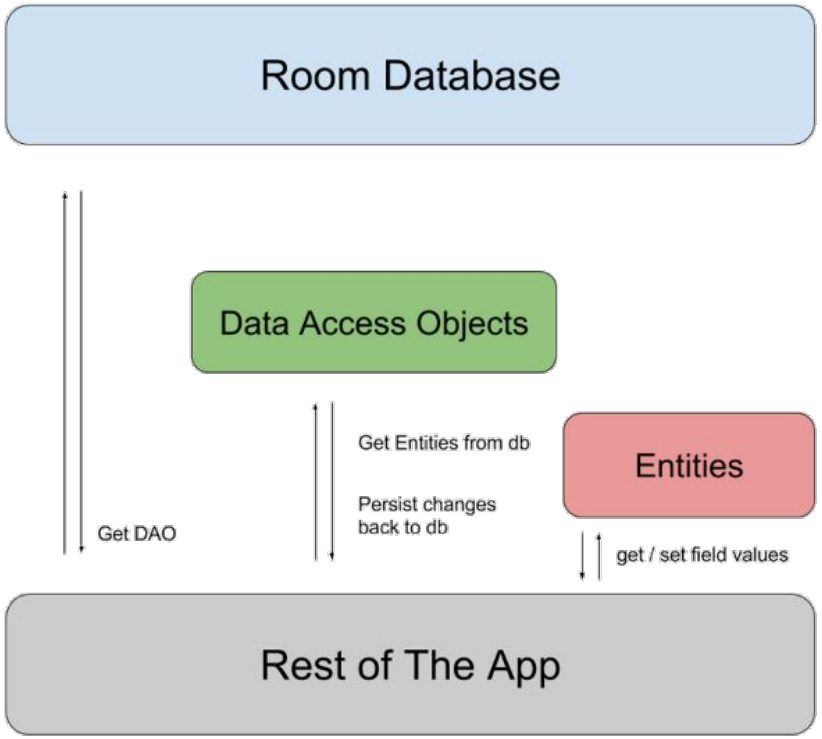
Storage

- When the app is killed, everything in memory disappear

- Three common way to store data :
  - PreferenceManager, to store primitive data (String, int, float)
  - Serialize data into a file
  - Use a SQL database or alternative (Realm)

Room

- ORM made by Google
- Provide an easy way to integrate an SQLite database storage into your app



Room

Demo

Misc

Service

- Background process (same as the app)
- Doesn’t display anything
- Can work while your app is in foreground or in background
- Service has been mostly replaced by JobScheduler (background tasks)
- Still used for foreground service, for bluetooth app or music
- Needs to have a notification since Android 8.0

Useful libraries

Android Jetpack

Bunch of libraries made by Google to provide solutions to common problem

- **Lifecycle** - Provide solutions to the complex lifecycle of activities and fragment
- **ViewModel** - MVVM architecture for Android App
- **LiveData** - observe data changes everywhere in your code
- **WorkManager** - background task execution on all devices
- **Room** - SQLite database ORM

Firebase

- Created in 2011, bought by Google in 2014
- Started as a NoSQL storage
- Now provide tens of services to servers, Android and iOS app.

Firebase

Some examples :

- **Storage** : Provide a remote database
- **Cloud Messaging** : Provide push and remote notification
- **Analytics** : Provide analytics insight of your app
- **Crash Reporting** : App crashs informations
- **Remote Config** : Remote configuration of app settings like a Summer/Winter mode for your app
- **Invites** : Provide URL to share your apps with parameters

Square

Square is a company that provide most of the greatest Android libraries

- **OkHttp** and Retrofit to make network calls
- **Picasso** to get image from URL and put them into ImageView
- **Timber** for logging
- Many more

Useful links

https://developer.android.com/docs/