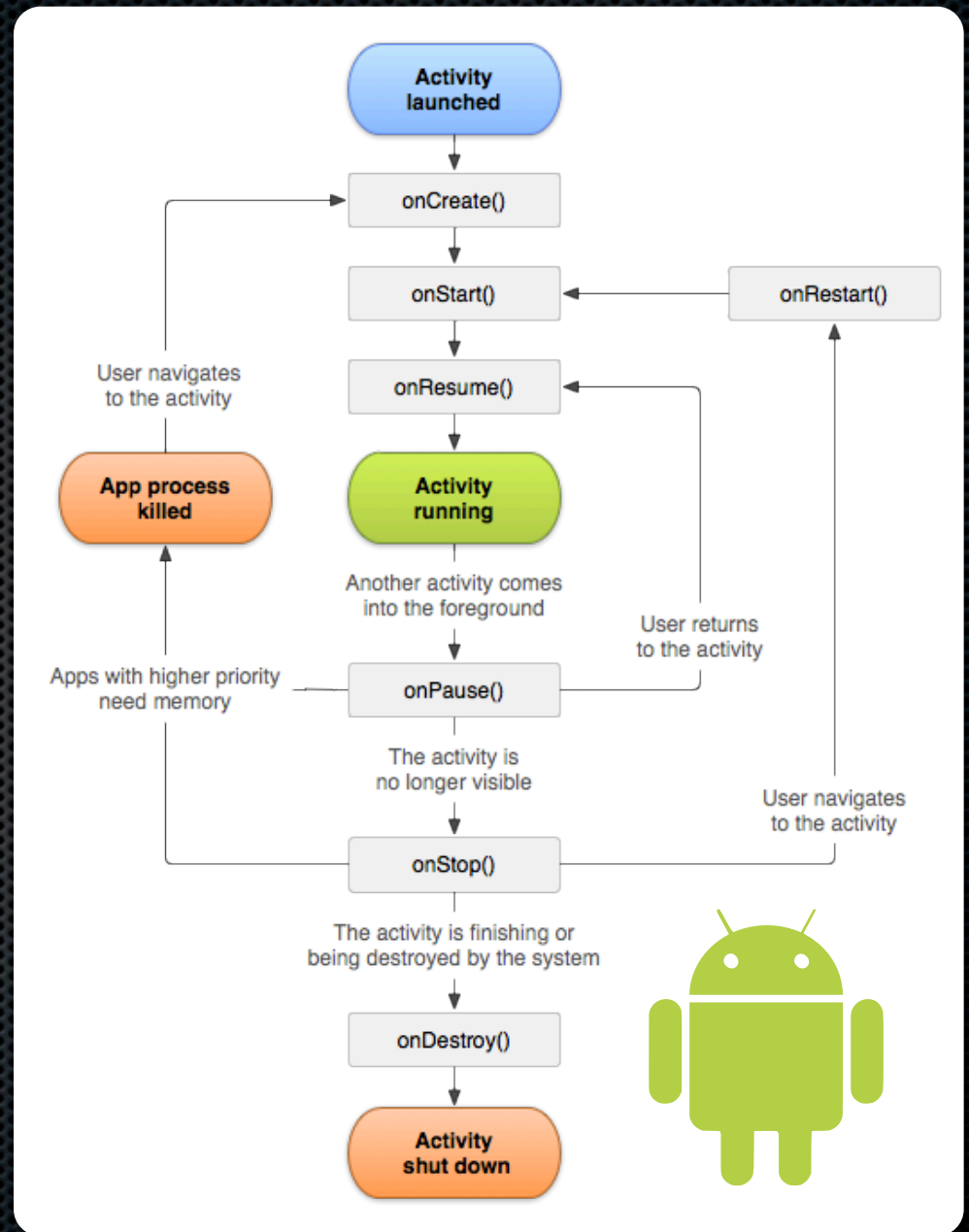


Mobile Application Programming: Android

Global Application State

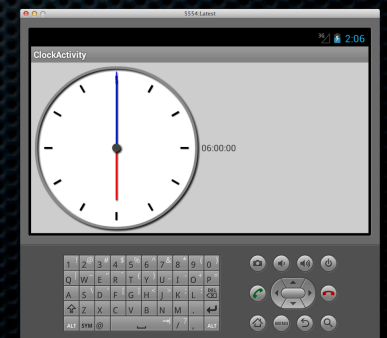
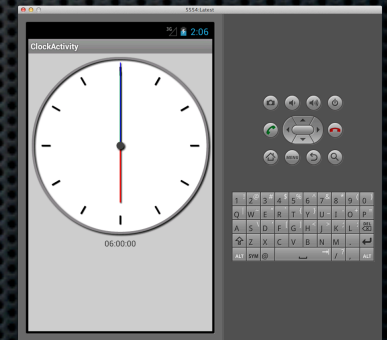
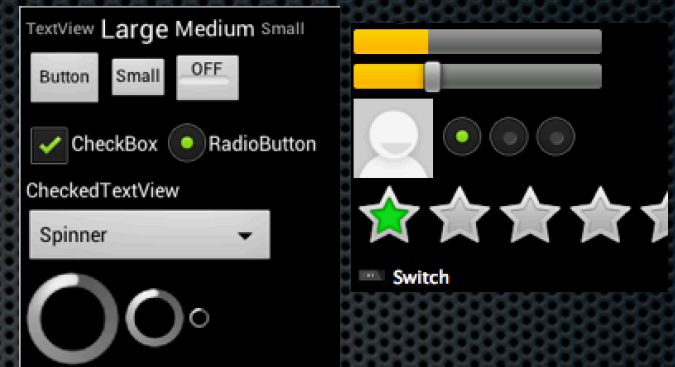
Activities

- ❖ Apps are **composed of activities**
- ❖ Activities are self-contained tasks made up of **one screen-full** of information
- ❖ Activities **start one another** and are **destroyed commonly**
- ❖ Apps can **use activities belonging to another app**



Custom Controls

- ✦ Create subclass of View class
- ✦ Override:
 - ✦ `onDraw(Canvas c)`
 - ✦ `onMeasure(int wMeasure, int hMeasure)`
- ✦ Add listener interface and listener property for the interesting events the control generates and call `on...` methods when events occur
- ✦ Implement view-level transient data saving



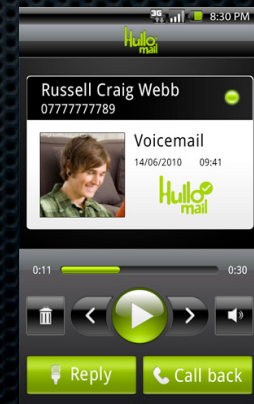
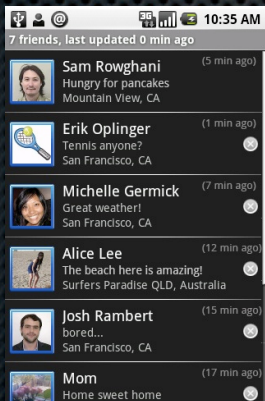
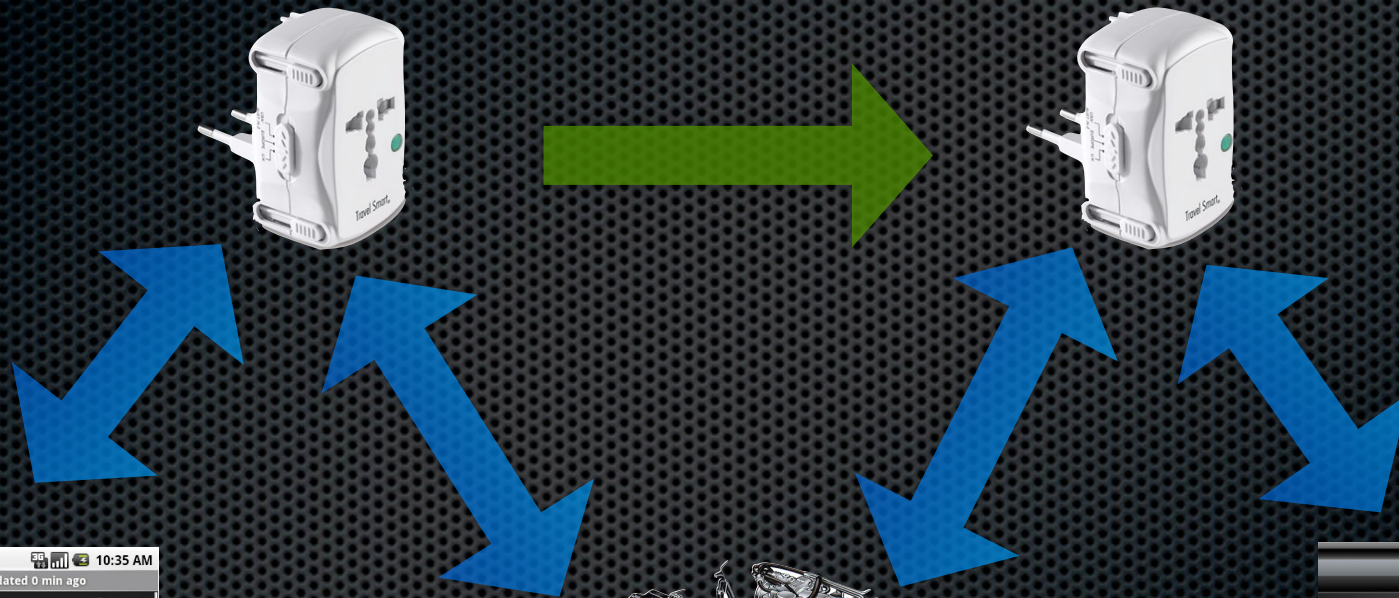
Application



Application Controller (MVC)

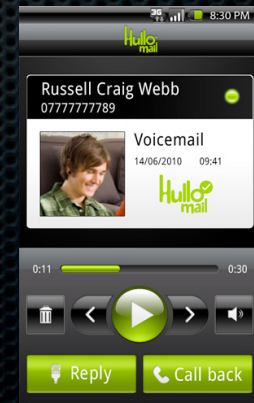
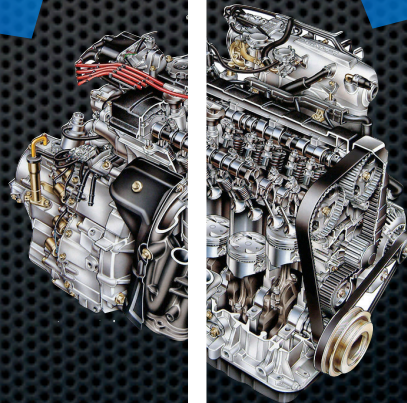
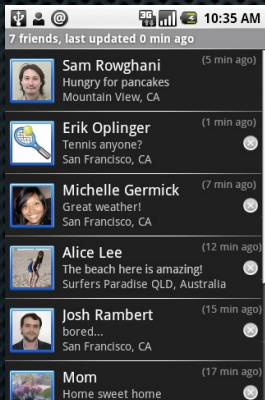
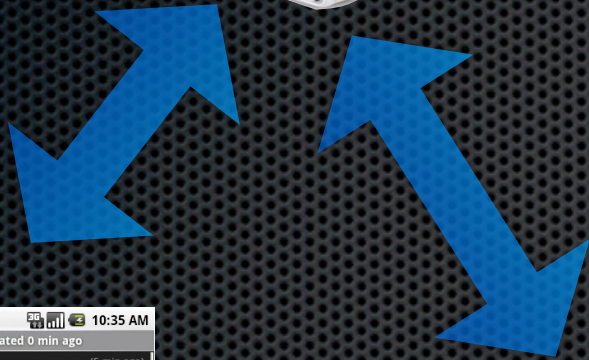


Launching Activities

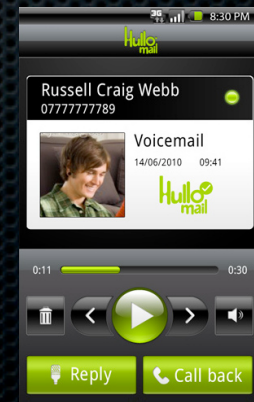
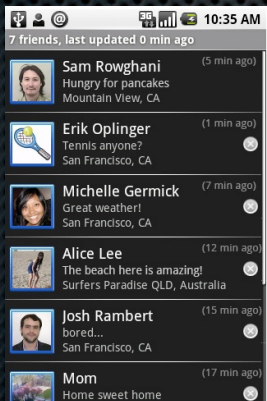
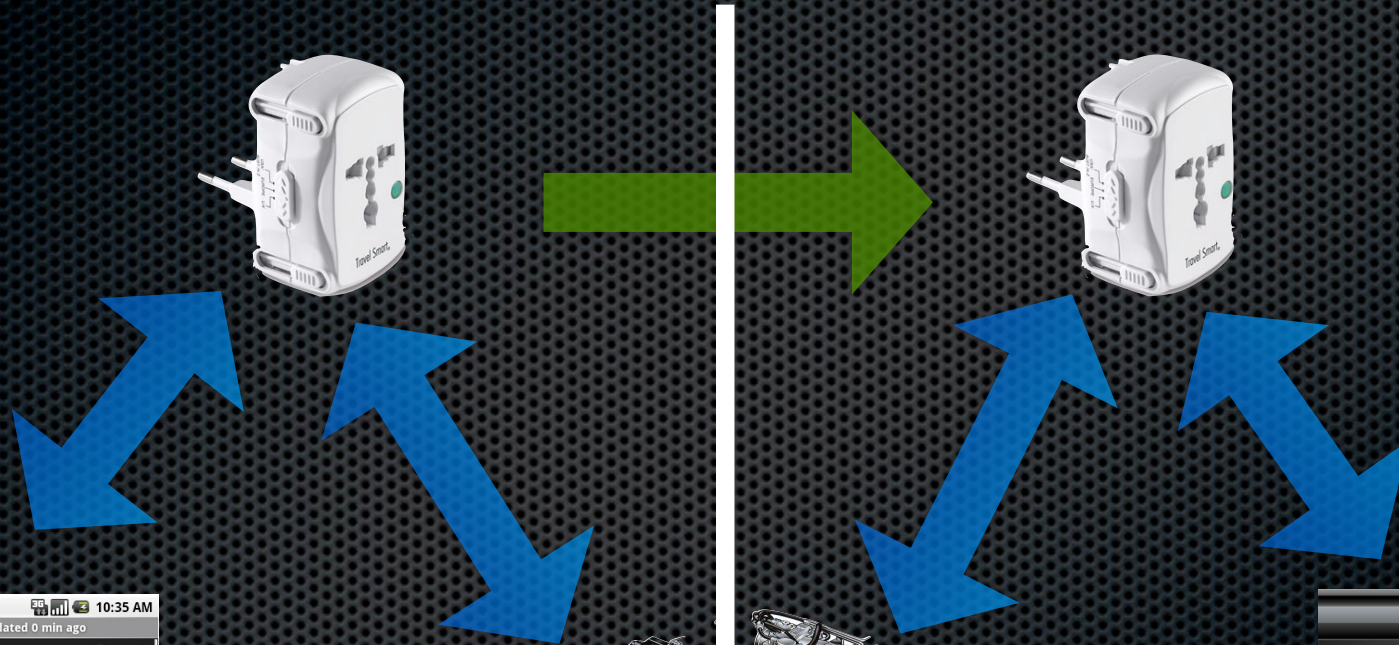


Launching Activities

Intent



Launching Activities



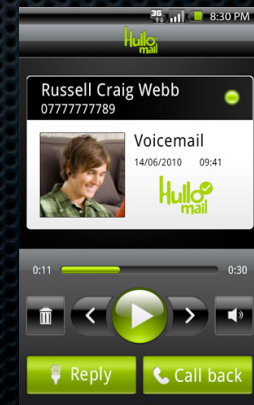
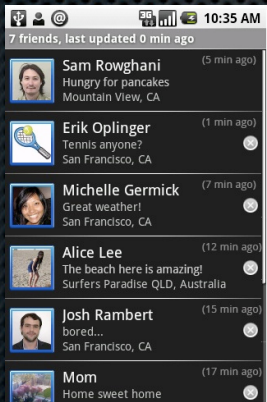
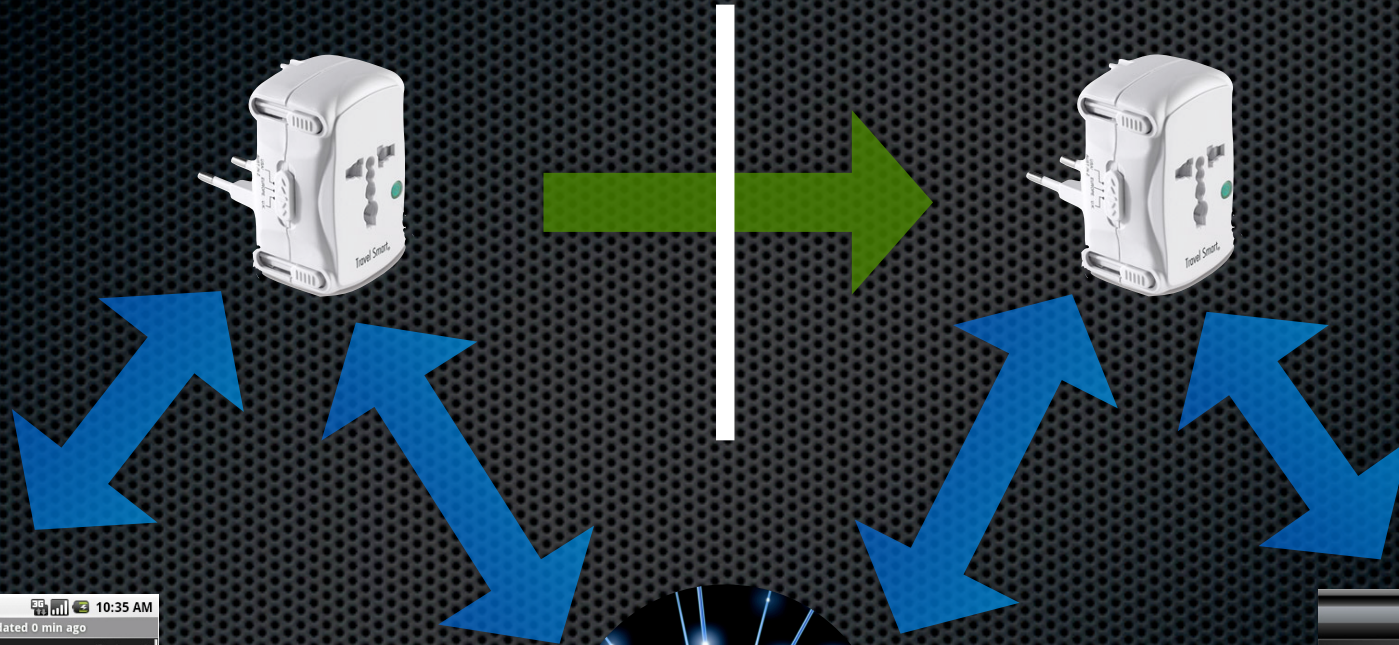
How to share this?

Application Class

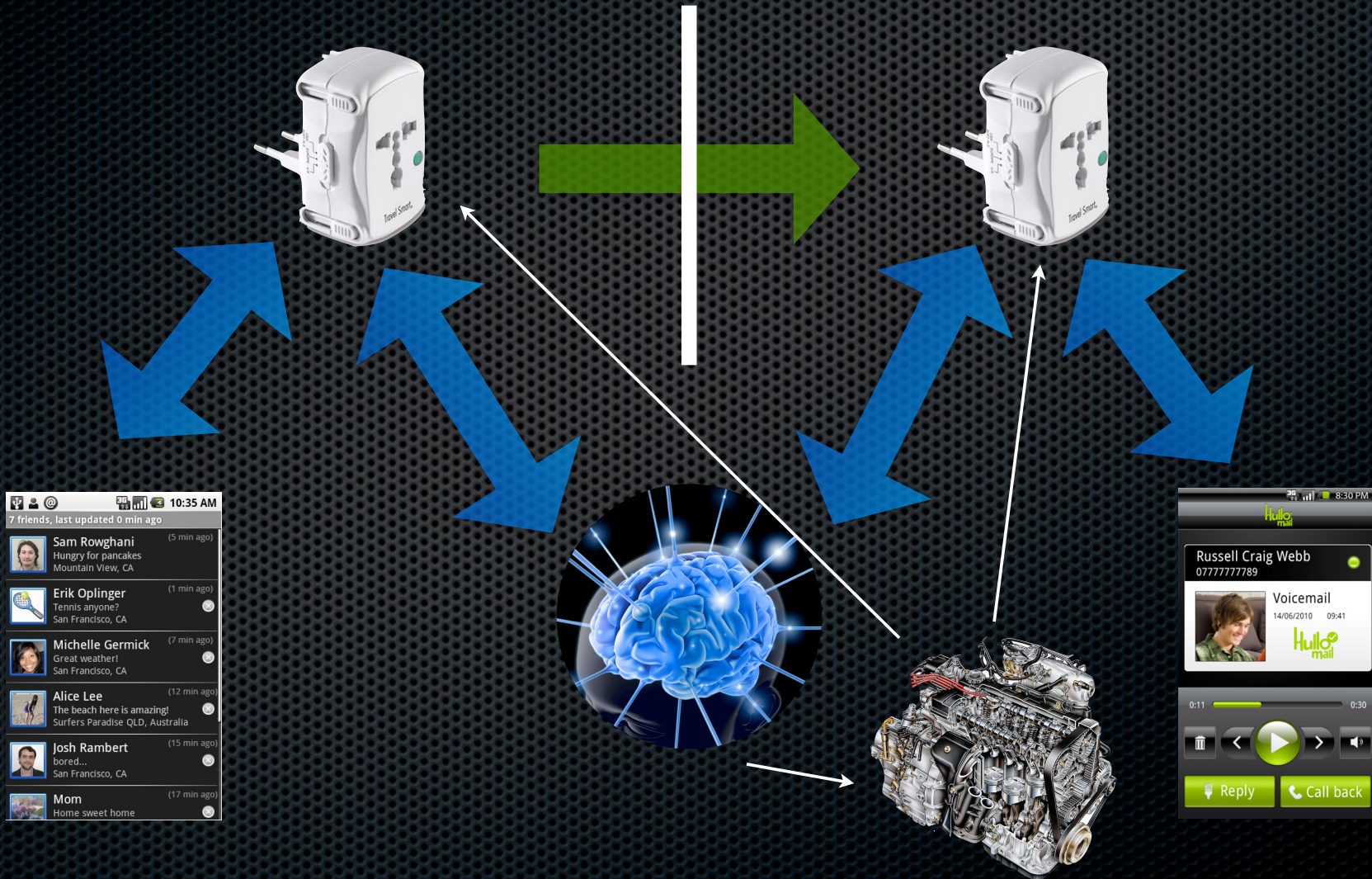


- ✦ Instantiated **before** any Activity
- ✦ Can be accessed using any Context object using **getApplication()** method
- ✦ Has an **onCreate** method
- ✦ Does **not** have an **onPause**
 - ✦ **onTerminate** also does not work as a replacement

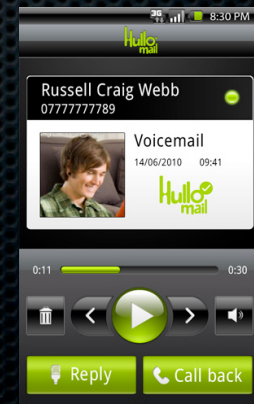
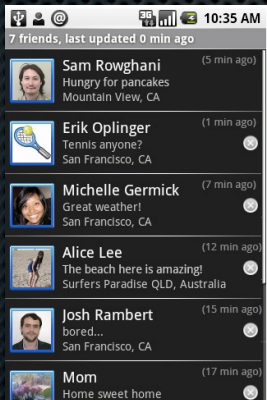
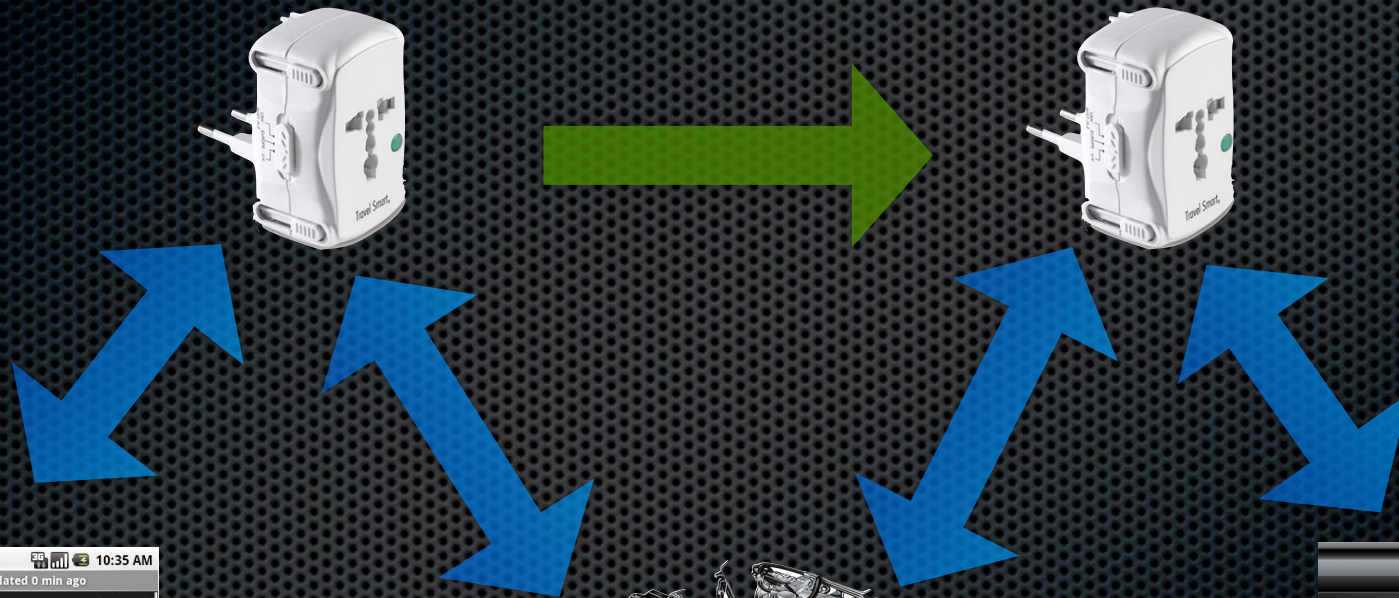
Sharing Using Application



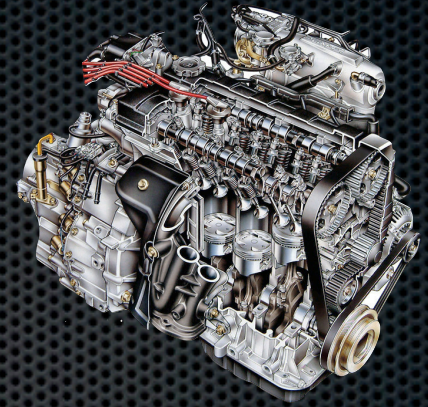
Sharing Using Application



Sharing Using Application



Singletons



- ✦ **Single-instance** class with **global state**
- ✦ Time-tested **design pattern**
- ✦ Enforces instantiation controls
- ✦ Has a **getInstance()** method that makes singleton
- ✦ Should be **thread-safe!**
- ✦ Need to define a way to make **persistent** and call in an appropriate place in your application