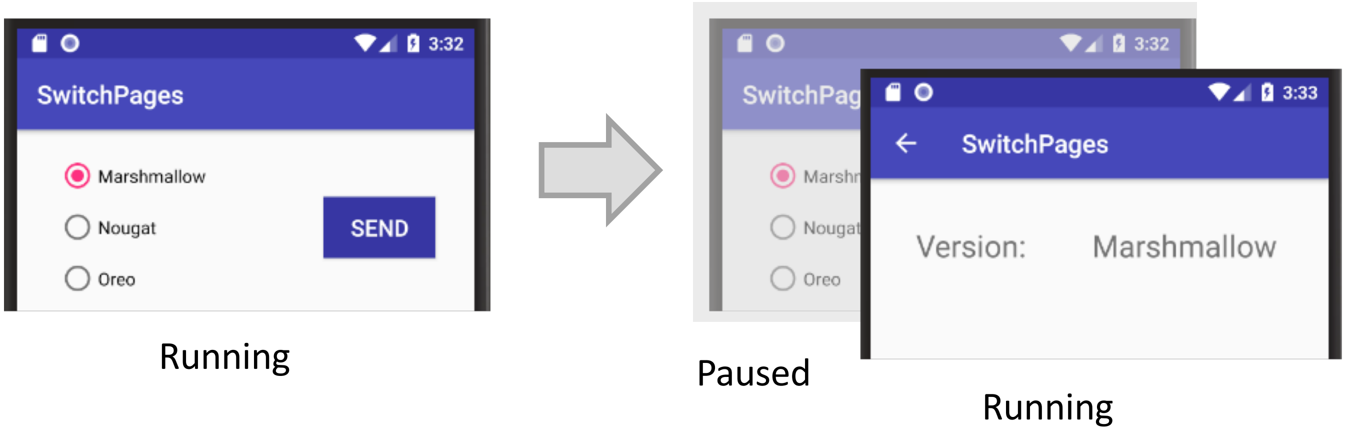
Multiple Views Application

Intents

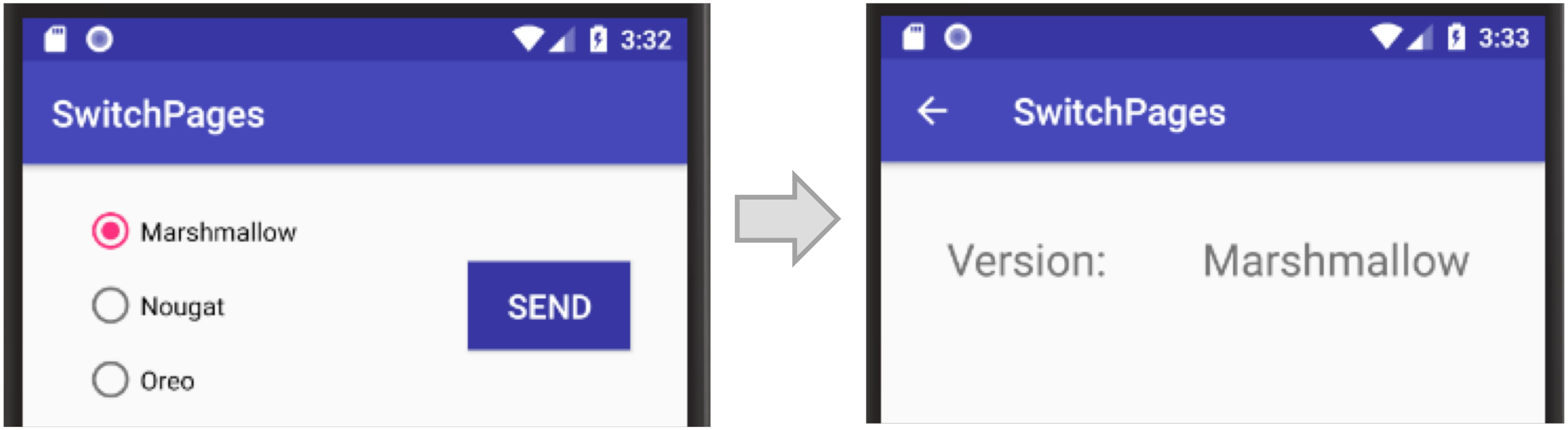
MVC in Android

Activity Lifecycle

- Usually, one view per activity
- Only one activity is running at any given time
- We often want to let users navigate between different views.
- How can we share data between views? Intents



Start Another Activity: Pass Data

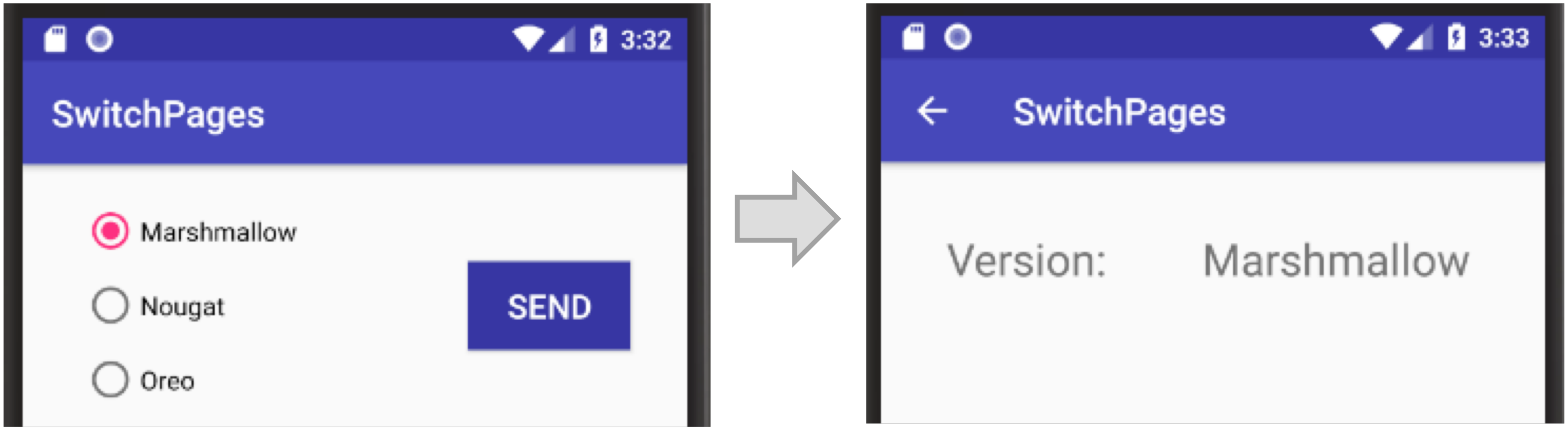


Version: Marshmallow

```
int version = 0;
switch (radio_id){
    case R.id.radioButton1:
        version = 6;
        break;
  …

Intent intent = new Intent(this, VersionActivity.class);
intent.putExtra("version", version); // data to pass
startActivity(intent);
```

Start Another Activity: Receive Data



Version: Marshmallow

```
// Get the Intent that started this activity
//and extract the value in int
Intent intent = getIntent();
int v = intent.getIntExtra("version",0);

// Set the string match to the value
TextView label = findViewById(R.id.version_txt);
 switch(v){
     case 6:
     label.setText(R.string.v6);
```

Return “Home” to Previous Activity

- In Manifest file, specify “parentActivity”

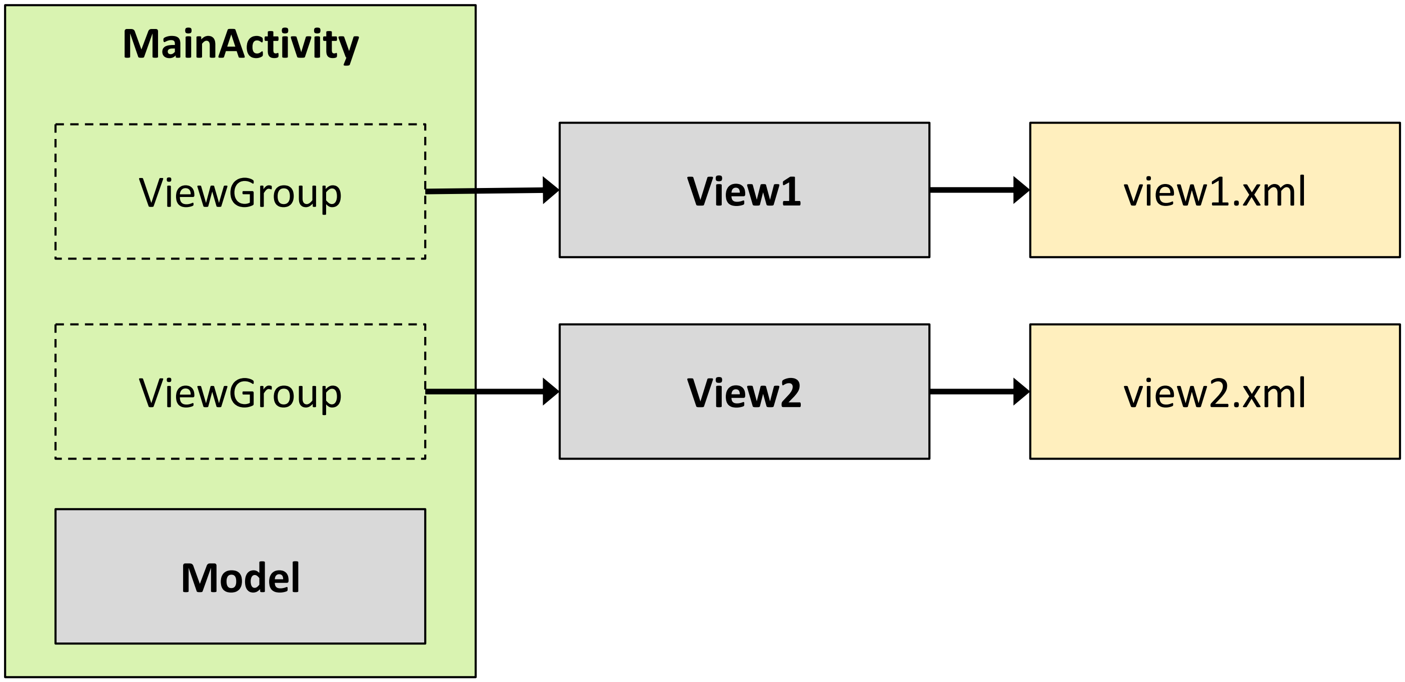
```
<application
    android:allowBackup="true”
        …
        >
    <activity android:name=".MainActivity">
        …
    </activity>

    <activity android:name=".VersionActivity"
              android:parentActivityName=".MainActivity”>
    </activity>
</application>
```

MVC in Android

- An Android UI is defined in XML
  - natural separation between the UI, application data and business logic
  - how to connect the view and the model??
- Google seems to recommend Model-View-ViewModel (MVVM)
  - view is connected to model through a view-model
  - the view-model manages data binding from view to model
- Other people use Model-View-Presenter (MVP)
  - where presenter is broker between view and model
- Both are suggesting a mediating layer between View and Model

MVC Structure Option 1



MainActivity

ViewGroup

ViewGroup

Model

View1

View2

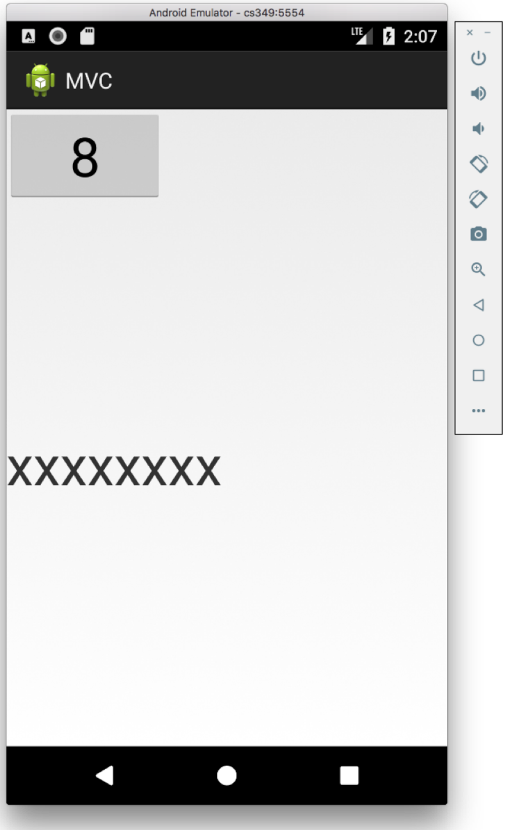
view1.xml

view2.xml

🟩 Activity 🟨 UI xml ⬜ Other Class

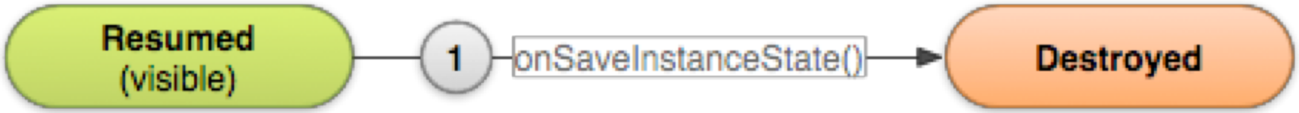
Code Demo: MVC1

- Views as Android ViewGroups
  - similar to desktop MVC we discussed earlier
- Notes:
  - Model essentially identical to desktop Java
  - “Inflating” view layouts into main view
  - `onPostCreate()` when inflating view layouts
  - Save and restore model

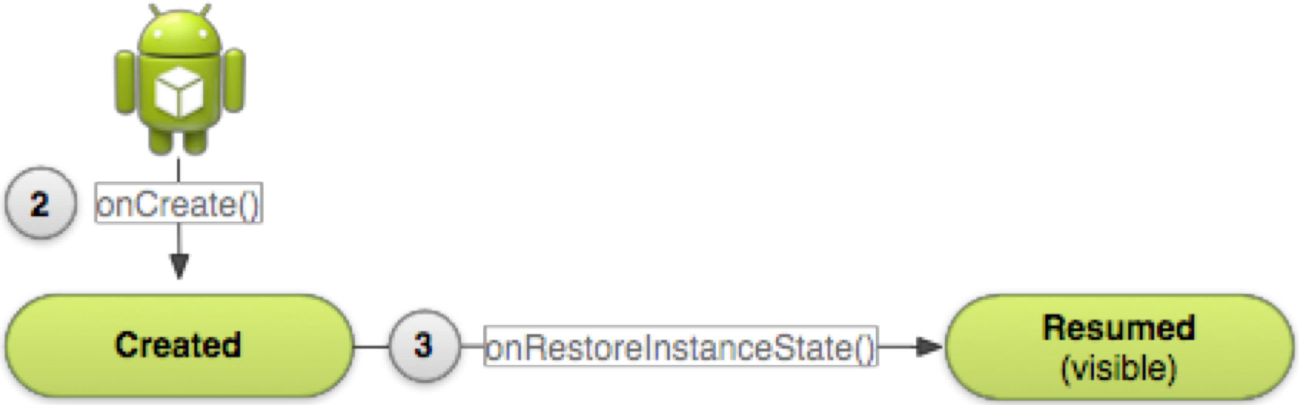


Recreating an Activity

Use built-in methods to save and restore model state.

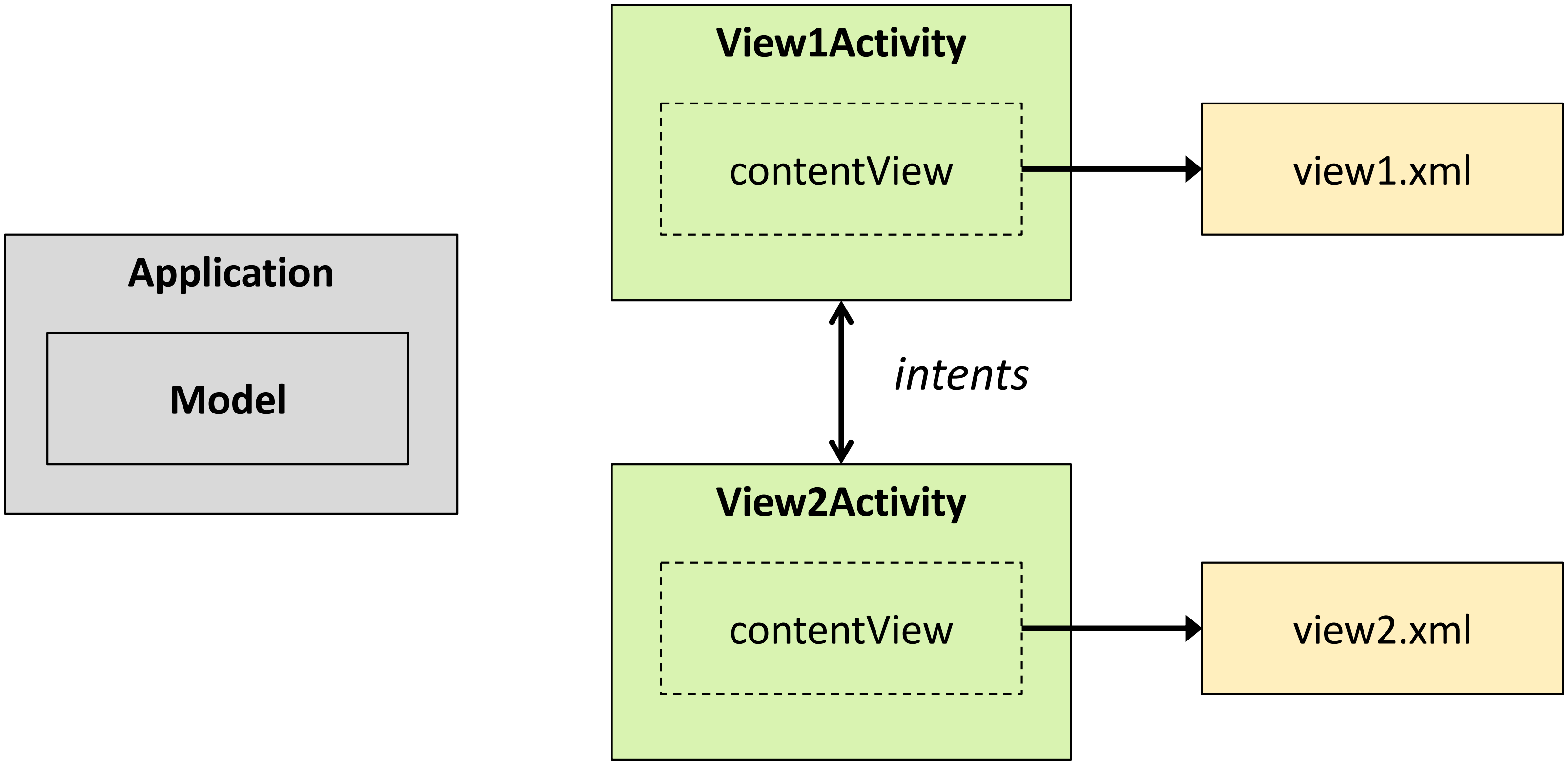


```
protected void onSaveInstanceState(Bundle outState) {
   outState.putInt("Counter", model.getCounterValue());
   ...
}
```



```
protected void onRestoreInstanceState(Bundle savedInstanceState) {
   model.setCounterValue(savedInstanceState.getInt("Counter"));
   ...
}
```

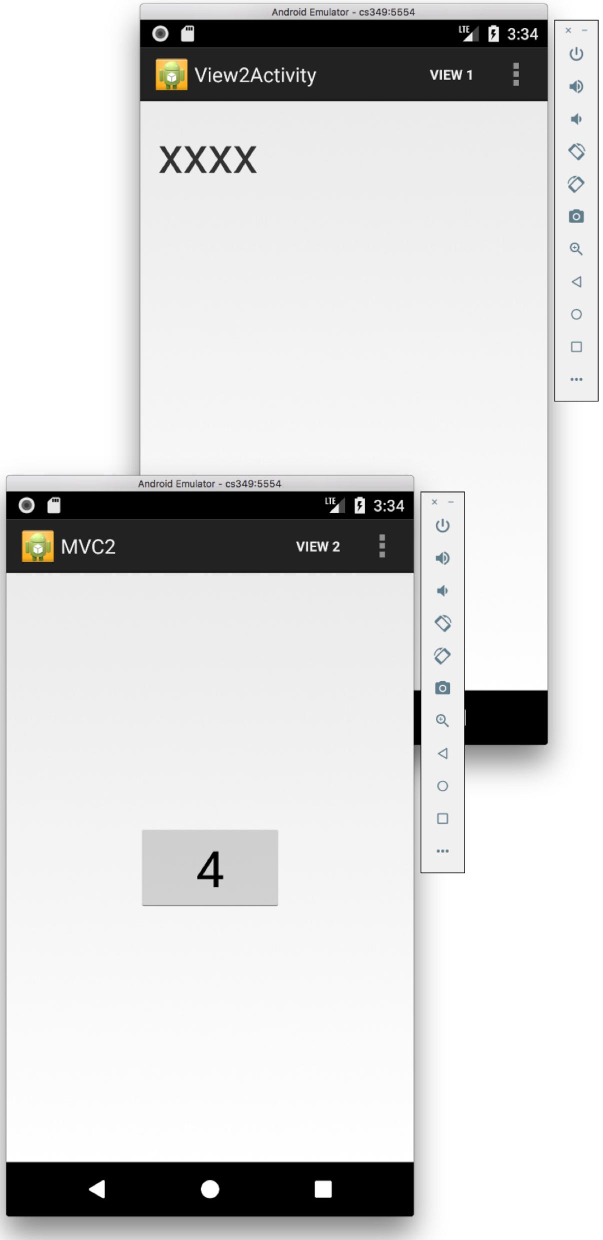
MVC Structure Option 2



🟩 Activity 🟨 UI xml ⬜ Other Class

Code Demo: MVC2

- Views as Activities, global static model
- Notes:
  - Application class
  - `onDestroy(), deleteObserver()`
  - Create options menu
  - Intents to start activity
  - `finish()`
  - no need to persist model!


Android Emulator - cs349:5554
3:34
View2Activity
VIEW 1
XXXX
Android Emulator - cs349:5554
3:34
MVC2
VIEW 2
4