

Recording in Progress

This class is being recorded

Please turn off your video and/or video if you do not wish to be recorded

CMSC436: Programming Handheld Systems

Permissions

Today's Topics

Android permissions

Defining and using permissions

Component permissions and related APIs

Permissions

Permissions protects resources and data

For instance, they limit access to:

- User information – e.g., Contacts

- Cost-sensitive API's – e.g., SMS/MMS

- System resources – e.g., Camera

Permissions

Permissions are represented as strings

Apps describe relevant permissions in
AndroidManifest.xml, including

- Permissions they use

- Permissions required of components that want to interact with them

Using Permissions

Applications specify permissions they use through a `<uses-permission>` tag

These permissions must be granted before access is allowed

Apps must check at runtime that all required permissions have been granted

Using Permissions

```
<manifest ... >  
...  
<uses-permission android:name="android.permission.CAMERA"/>  
<uses-permission android:name="android.permission.INTERNET"/>  
<uses-permission  
    android:name="android.permission.ACCESS_FINE_LOCATION"/>  
...  
</manifest >
```

See: <https://developer.android.com/training/permissions/index.html>

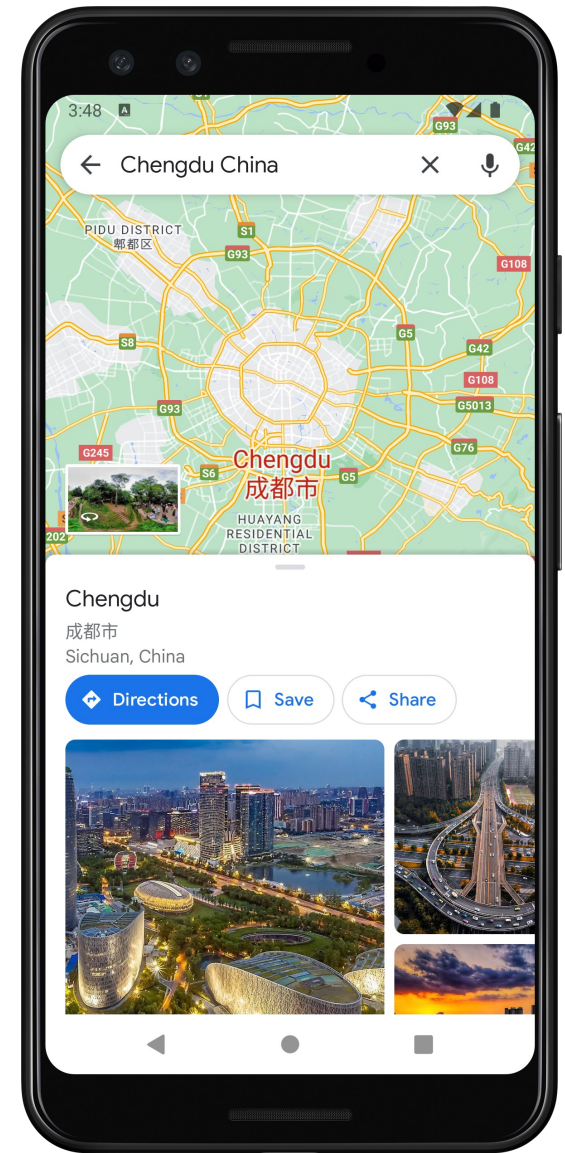
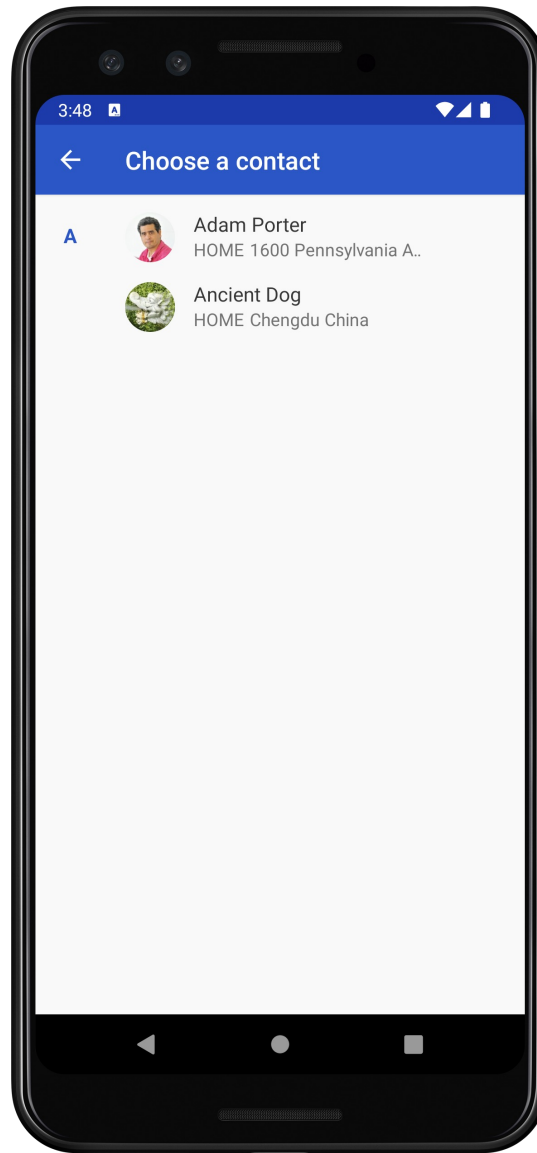
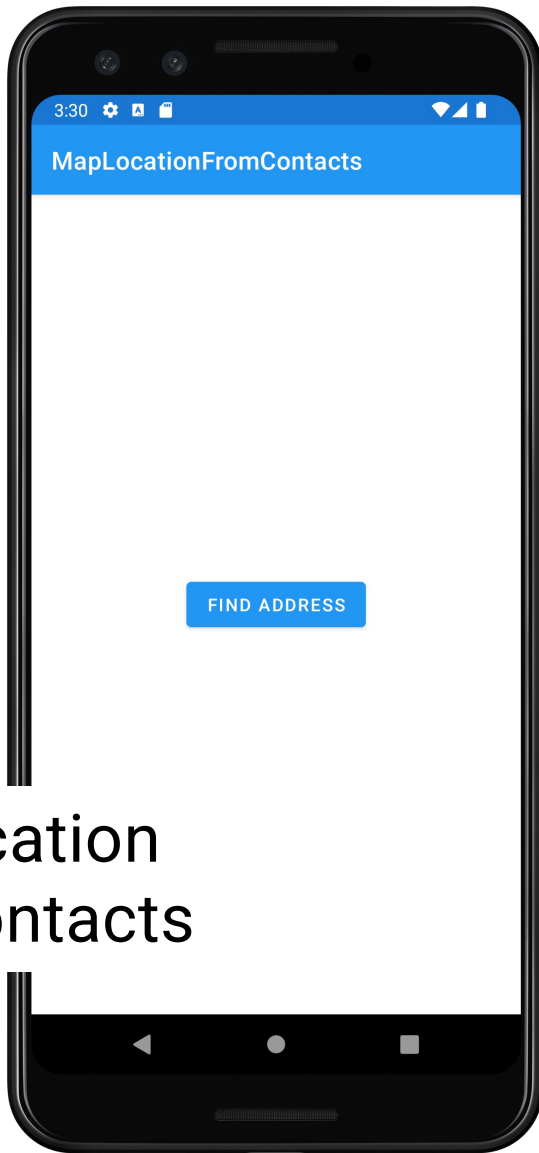
MapLocationFromContacts

Selects a contact from contacts database

Displays a map centered on selected contact's address

Requires permissions to read user's contacts db

MapLocation FromContacts



Permission Granting Recipe

1. Check whether permissions have been granted
2. If not, check whether user has previously denied permission
 1. Can try to explain the need for this permission and possibly ask user to grant permission
3. If user has not been asked for permission, do so now

See: `MapLocationFromContacts`

ActivityResultCaller.registerForActivityResult()

Example use case

Define `ActivityResultLauncher<String>` instance

This instance calls `registerForActivityResult()`, passing in necessary callback info

This info involves `requestPermission()` or `requestMultiplePermissions()` contract interface object

Call `ActivityResultLauncher<String>.launch(String)` to request desired permissions

Registered callback is started when permission request returns

Defining Permissions

Apps can also define and enforce their own permissions

Defining Permissions

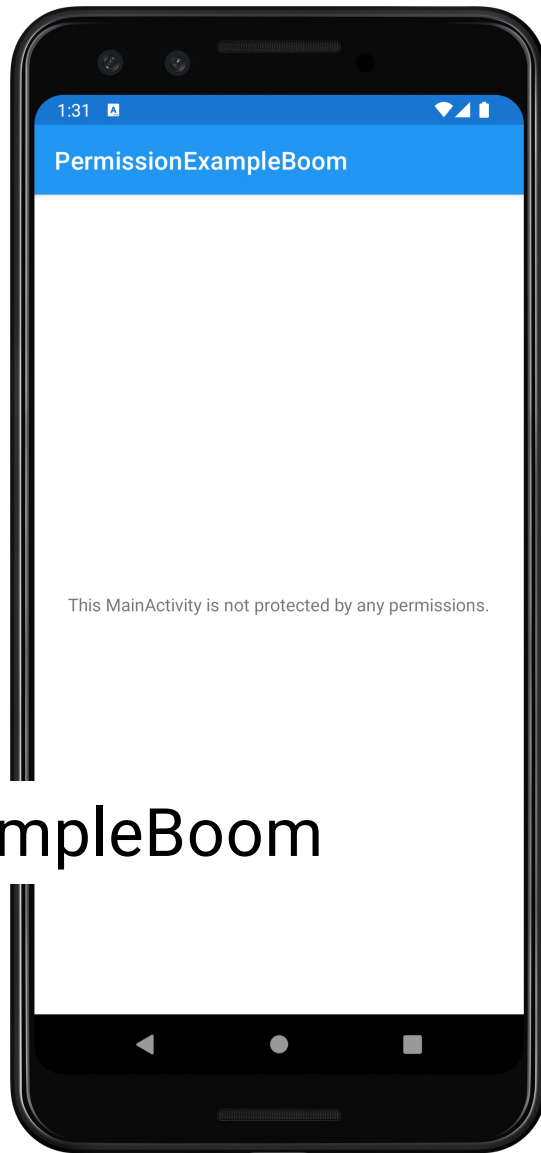
Suppose your application performs a potentially dangerous operation

You might not want to allow just any application to invoke yours

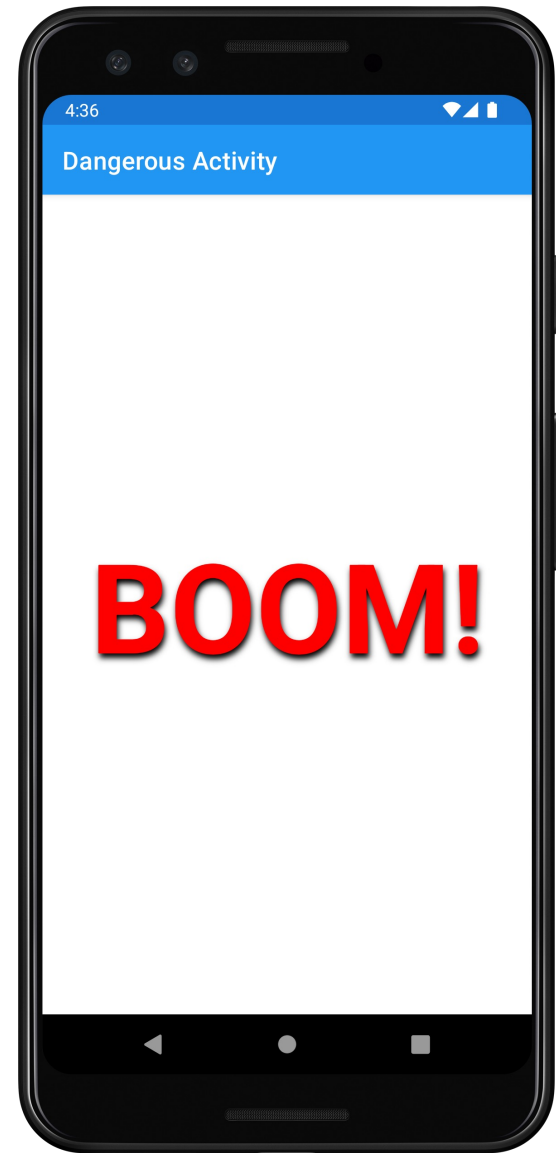
Android lets you define & enforce your own permissions

PermissionExampleBoom

Simple Application that performs a (pretend)
dangerous action



PermissionExampleBoom



Define & Enforcing Permissions

You don't want just any application to run
PermissionExampleBoom

Define & enforce an application-specific
permission

AndroidManifest.xml

```
<!-- Defines a custom permission -->
<permission
    android:name="course.examples.permissionexample.BOOM_PERM"
    android:description="@string/boom_perm_string"
    android:label="@string/boom_permission_label_string"
    android:protectionLevel="dangerous" />

<!--
Enforces the BOOM_PERM permission on users of this application
-->
<application
    android:allowBackup="false"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:permission="course.examples.permissionexample.BOOM_PERM"
    android:theme="@style/MaterialTheme">
    ...
```

ProtectionLevel

Normal – Low risk

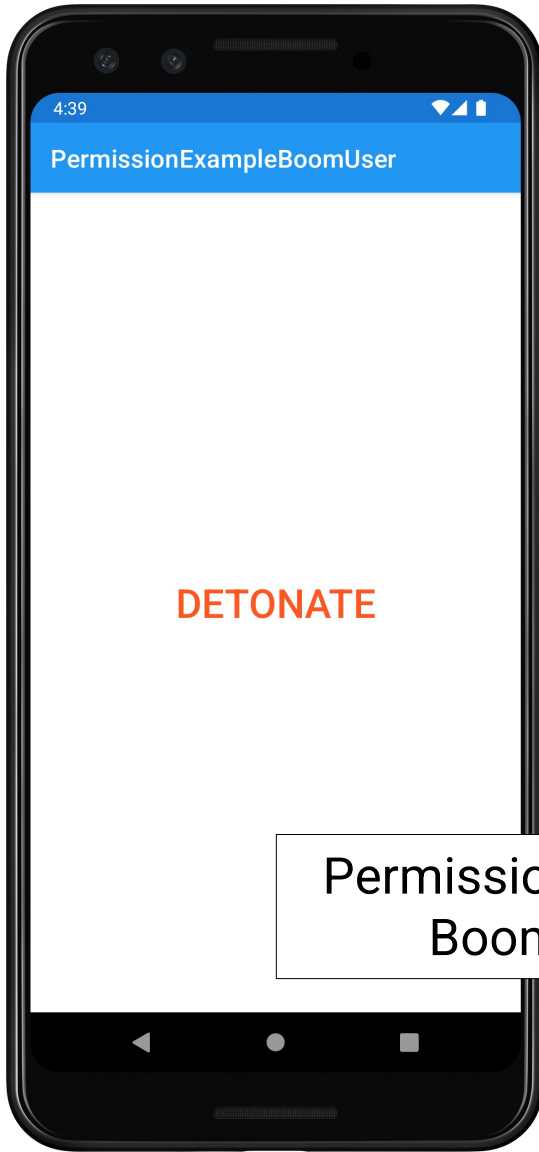
System automatically grants permission

Dangerous– High risk

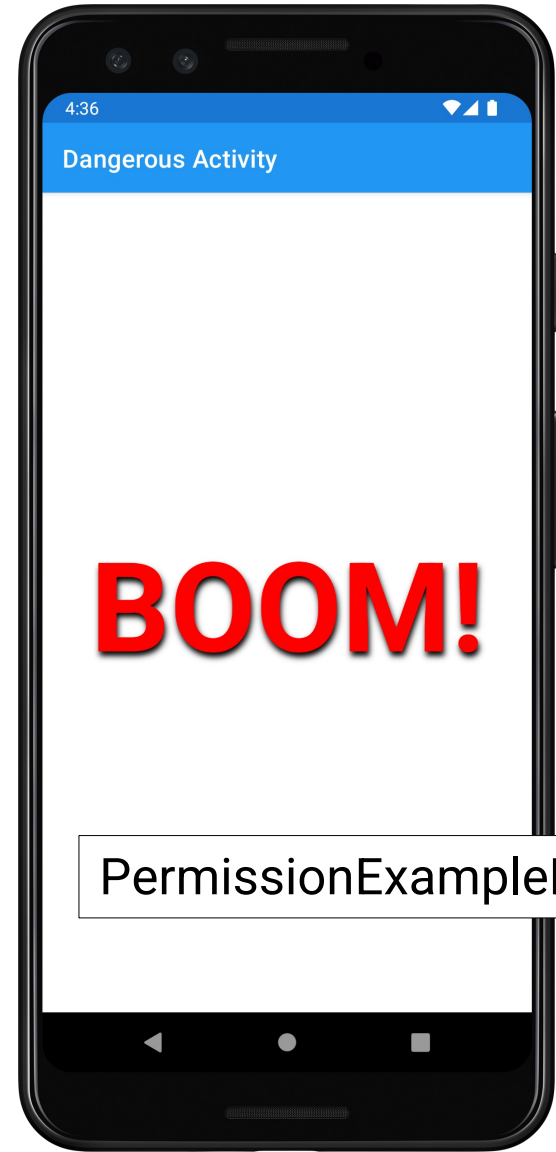
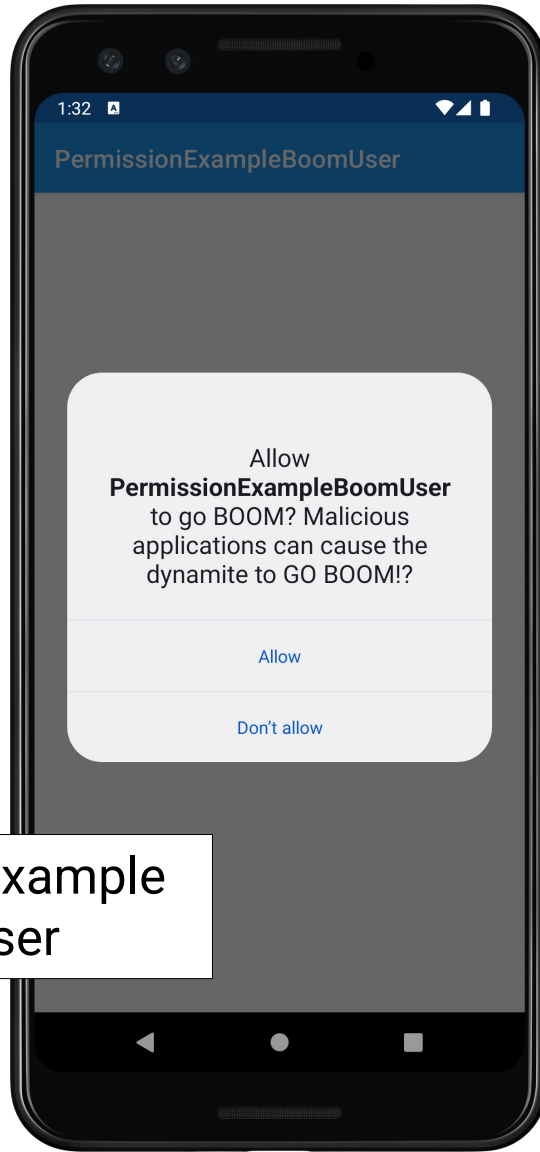
User must explicitly grant permission

Using the Permission

Apps that want to use `PermissionExampleBoom` must acquire the required permission



PermissionExample
BoomUser



PermissionExampleBoom

Uses-Permission

Application declares that it needs permissions required by other Applications it uses

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="course.examples.permissionexample.boomuser"
    android:versionCode="1"
    android:versionName="1.0" >

    <!-- App needs the "...BOOM_PERM permission -->
    <uses-permission
        android:name="course.examples.permissionexample.BOOM_PERM" />
```

Component Permissions

Individual components can set their own permissions, restricting which other components can access them

Component permissions take precedence over application-level permissions

Activity Permissions

Restricts which components can start the associated Activity

Checked within execution of

`startActivity()`

`startActivityForResult()`

Throws `SecurityException` on permissions failure

Service Permissions

Restricts which components can start or bind to the associated service

Checked within execution of

`Context.startService()`

`Context.stopService()`

`Context.bindService()`

Throws `SecurityException` on permissions failure

BroadcastReceiver Permissions

Restricts which components can send & receive broadcasts

Permissions checked in multiple places

More on this when we discuss
BroadcastReceivers

ContentProvider Permissions

Restrict which components can read & write the data in a ContentProvider

More on this when we discuss ContentProviders

Next

DataManagement

Example Applications

MapLocationFromContacts

PermissionBoom

PermissionBoomUser