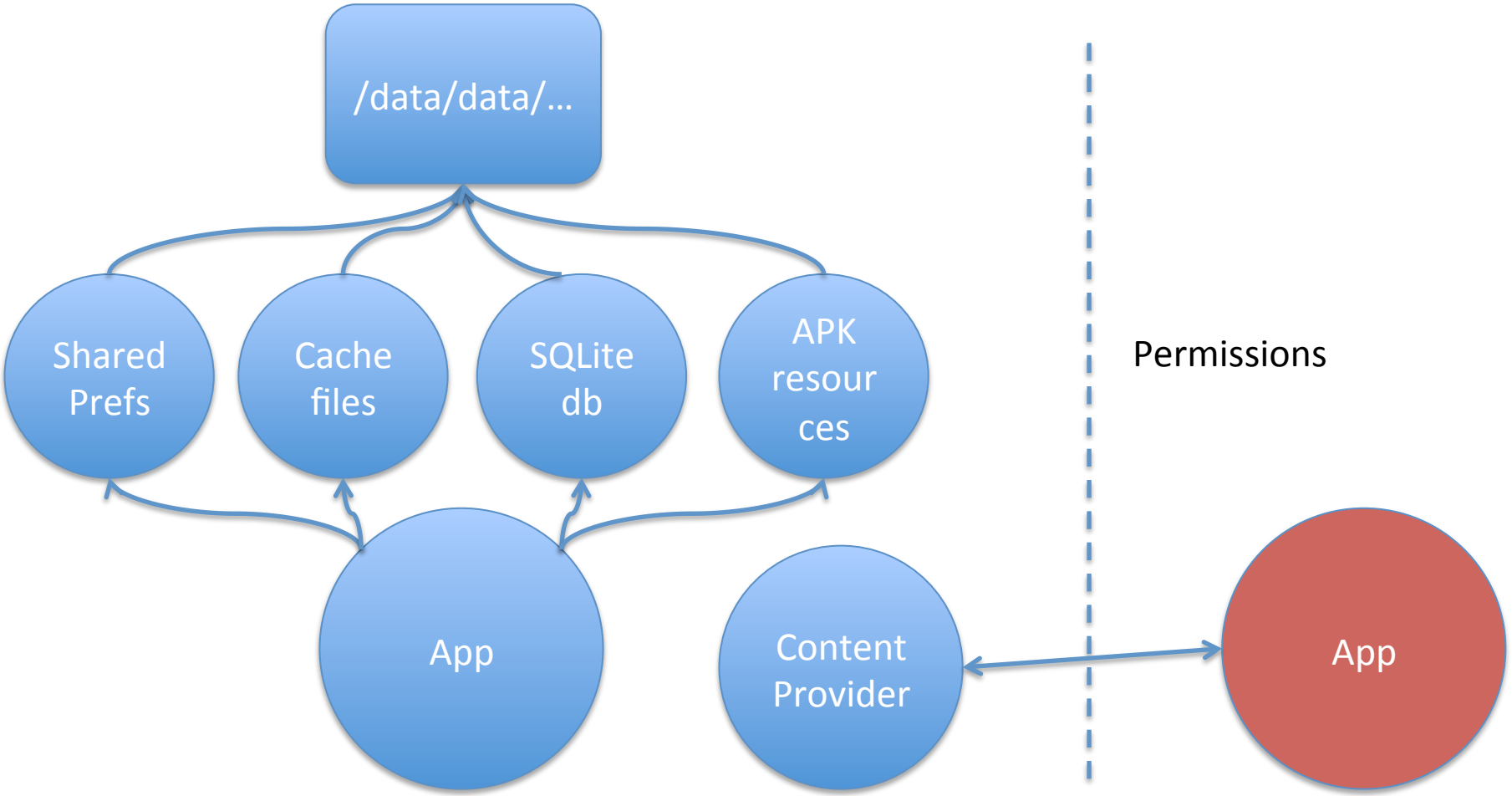
G54MDP
Mobile Device Programming

Lecture 12 – ContentProviders, Permissions and Security, BroadcastReceivers

git

- http://git-scm.com/
- https://github.com/mdf/g54mdp
- git clone https://github.com/mdf/g54mdp.git
- git pull

Sharing Data – if not



Creating a Content Provider

- Implement a storage system for the data
  - Structured data / SQLite
    - Values, binary blobs up to 64k
    - Database
  - Large binary blobs
    - Files
  - Photos / media manager
- Implement a ContentProvider
  - Implement required methods
  - query, add, update, insert etc
  - onCreate
  - getType
    - What type of data are we providing?
  - ParcelFileDescripter openFile()
- Tell Android we are a provider
  - Declare in the AndroidManifest

Contract

- Defines metadata pertaining to the provider
- Constant definitions that are exposed to developers via a compiled .jar file
  - Authority
    - Which app is responsible for this data
  - URI
  - Meta-data types
  - Column names
    - Abstraction of database architecture

URI Matching

- All of these methods (except onCreate()) take a URI as the first parameter
  - The object will need to parse it to some extent to know what to return, insert or update
  - Android provides android.content.UriMatcher to simplify this
    - Provides mapping between abstraction of contract class to concrete db implementation
  - Does the calling application want all data from a table, or just a row, or a specific table?
    - Or a “virtual” table

Let’s have a look…

Network

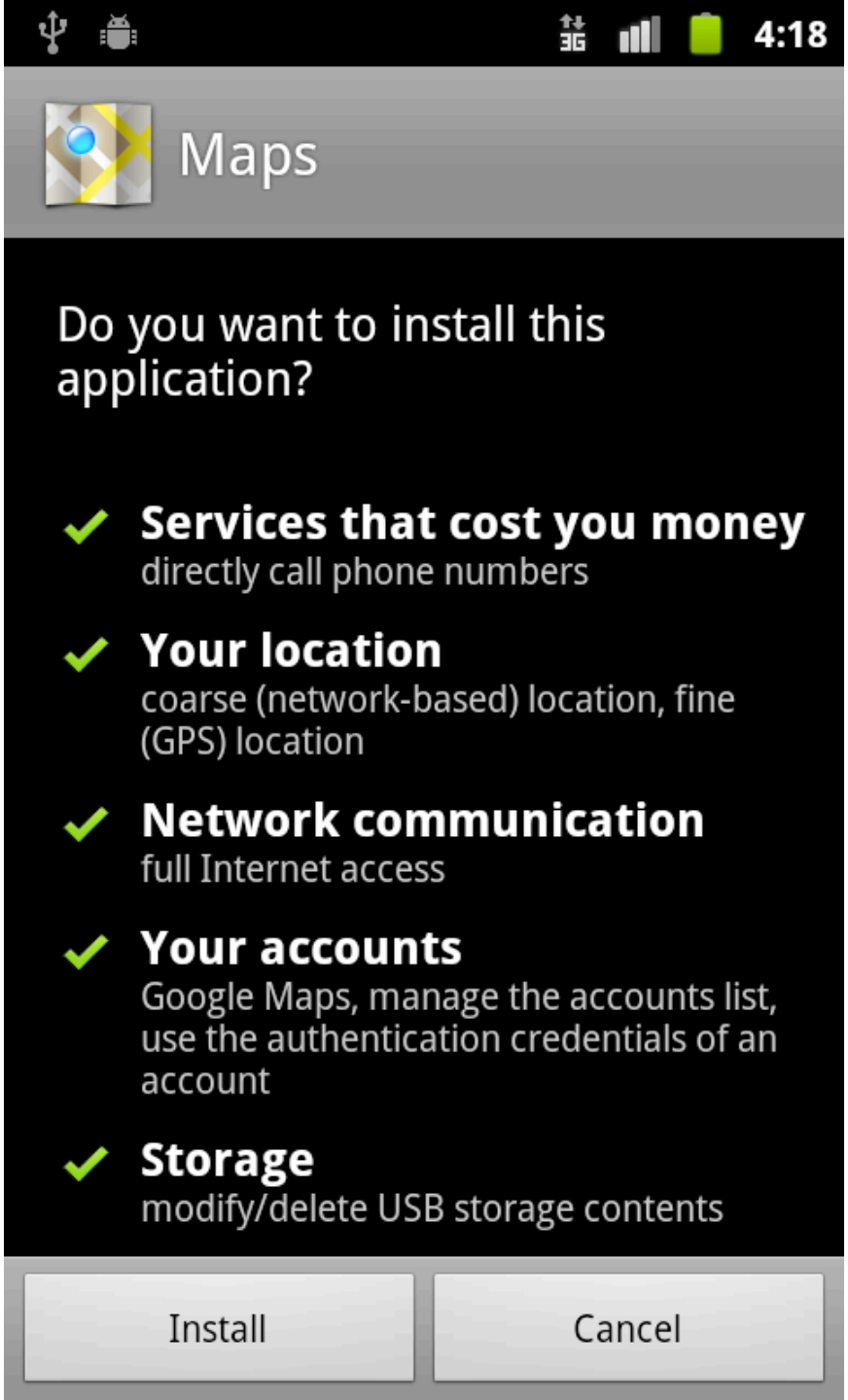
- One last type of data storage
  - Get it off the phone, and into the cloud
- Implement a SyncAdapter
  - Appears in the “Accounts and Sync” menu in the OS
  - Synchronizes a local database / content provider with a remote server
  - Make use of a Service to push data in the background
- http://developer.android.com/training/sync-adapters/creating-sync-adapter.html

Android Security

- Isolation by default
- Linux kernel
  - Filesystem / UID
    - Private, per-application file storage
  - Processes
    - Individual virtual machine instances
    - Native-code controlled by the application sandbox
- Restricted access to the root user
  - Most processes run as normal users
- IPC through specific interfaces
  - Services, Binders, Intents, Messages, ContentProviders
  - Intentional lack of APIs for sensitive functionality
    - Direct SIM card access

Permissions

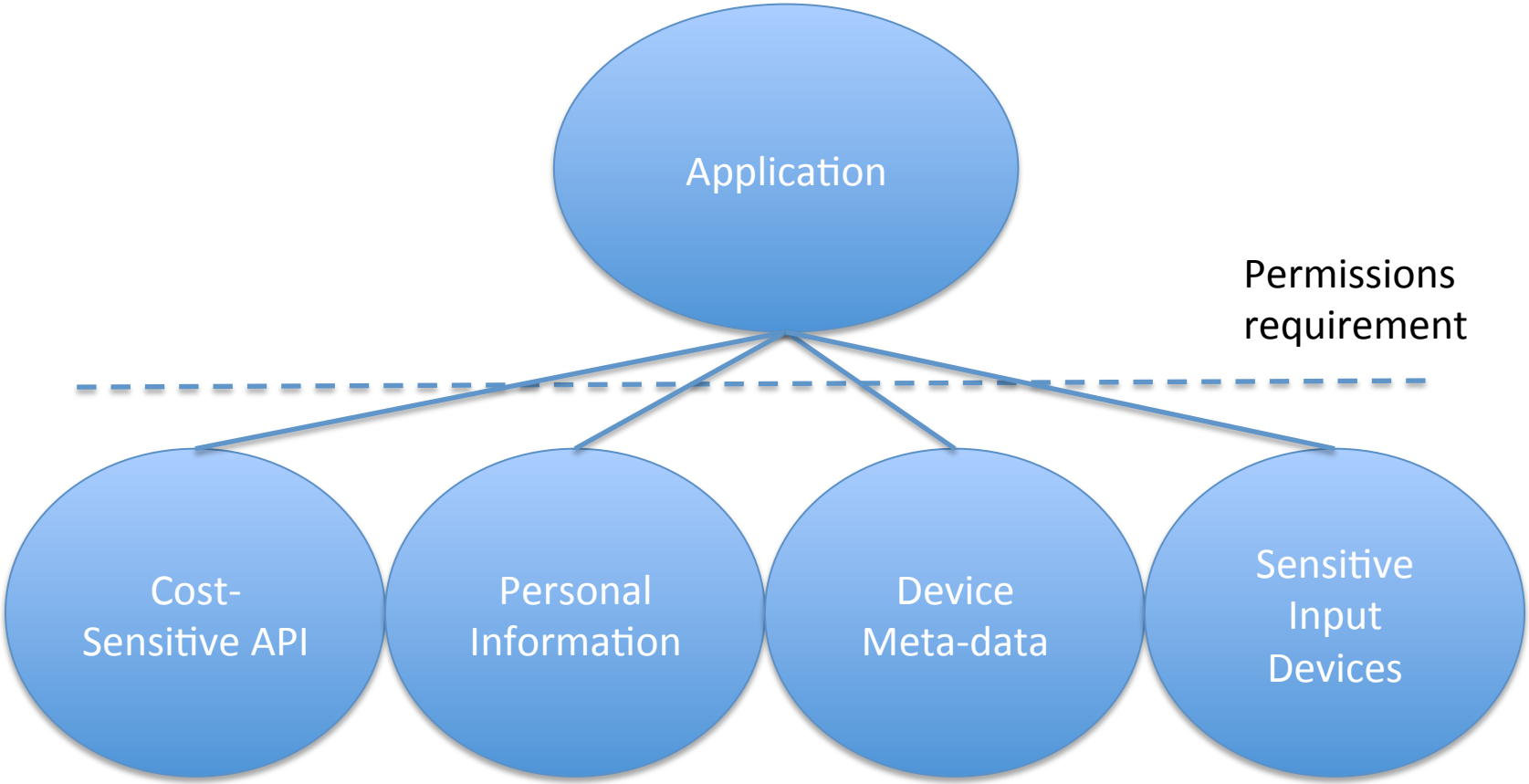
- No access by default
  - Control access to specific mechanisms
- Applications can offer protected access to resources and data with **permissions**
  - Permissions explicitly granted by users
- Permission architecture
  - Applications statically declare permissions
    - Required **of** components interacting with them
      - You must have this permission to interact with me
    - Required **by** components they interact wth
      - I will need these permissions
  - Android requires user's consent to specific permissions when an application is installed


4:18
Maps
Do you want to install this application?
Services that cost you money
directly call phone numbers
Your location
coarse (network-based) location, fine (GPS) location
Network communication
full Internet access
Your accounts
Google Maps, manage the accounts list, use the authentication credentials of an account
Storage
modify/delete USB storage contents
Install
Cancel

Permissions

- Show permissions required at install time
  - Not prompted again regarding permissions at run-time
- Why?
  - Not yet made a commitment (financial, mental) to the application
    - Can compare to other applications
  - Not per session / at run-time
    - "Seamless" switching between Activities / applications
    - Would slow down the user experience
    - Train users to click "ok" repeatedly without considering the implications

Permissions



Permissions

- Cost-Sensitive APIs
  - Telephony
  - SMS/MMS
  - Network/Data
  - In-App Billing
  - NFC Access
- Personal Information
  - Contacts, calendar, messages, emails
- Device Meta-data
  - System logs, browser history, network identifiers
- Sensitive Input Devices
  - Interaction with the surrounding environment
  - Camera, microphone, GPS

Permissions

- http://developer.android.com/reference/android/Manifest.permission.html

String	PROCESS_OUTGOING_CALLS	Allows an application to modify or abort outgoing
String	READ_CALENDAR	Allows an application to read the user's calendar
String	READ_CALL_LOG	Allows an application to read the user's call log.
String	READ_CONTACTS	Allows an application to read the user's contacts
String	READ_EXTERNAL_STORAGE	Allows an application to read from external storag
String	READ_FRAME_BUFFER	Allows an application to take screen shots and m the frame buffer data.
String	READ_HISTORY_BOOKMARKS	Allows an application to read (but not write) the u bookmarks.

Common Permissions

- android.permission.ACCESS_FINE_LOCATION
- android.permission.WRITE_EXTERNAL_STORAGE
- android.permission.INTERNET
- android.permission.WAKE_LOCK

Using Permissions

- Applications can define new permissions in the manifest

  <permission android:name="android.permission.VIBRATE"

  …

  </>

  - Do we really need a new permission?
  - normal / dangerous / signed
  - "Readable" explanation of the new permission
- Applications can require components interacting with them to have a specified permission, set in the manifest
  - By default all permissions apply to all components hosted by the application
    - Activities, Services etc.
  - Or per component permission requirements

Using Permissions

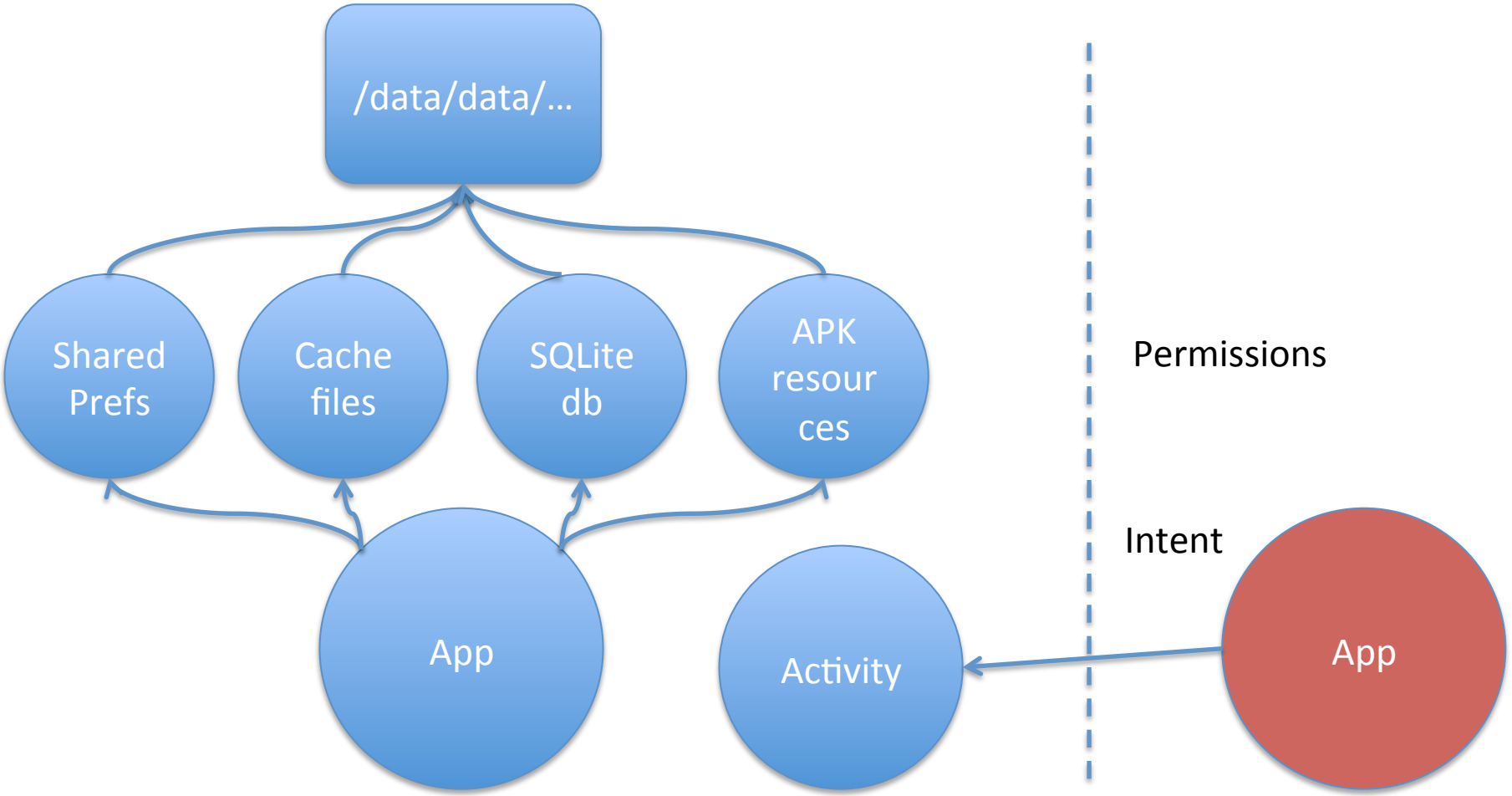
- Specify that an Application **uses** a permission

```
<uses-permission android:name="android.permission.CALL_PHONE" />
```

- Specify that an Application **requires** a permission
  - The app must **use** permissions it **requires**

```
<provider
    android:permission="android.permission.READ_CONTACTS”
    android:authorities="com.example.martincontentprovider.MyProvider”
    android:multiprocess="true”
    android:name="com.example.martincontentprovider.MyProvider">
</provider>
```

Sharing Data – is this good enough?



References

- http://developer.android.com/guide/topics/providers/content-providers.html
- http://developer.android.com/guide/components/fundamentals.html
- http://developer.android.com/reference/android/content/BroadcastReceiver.html