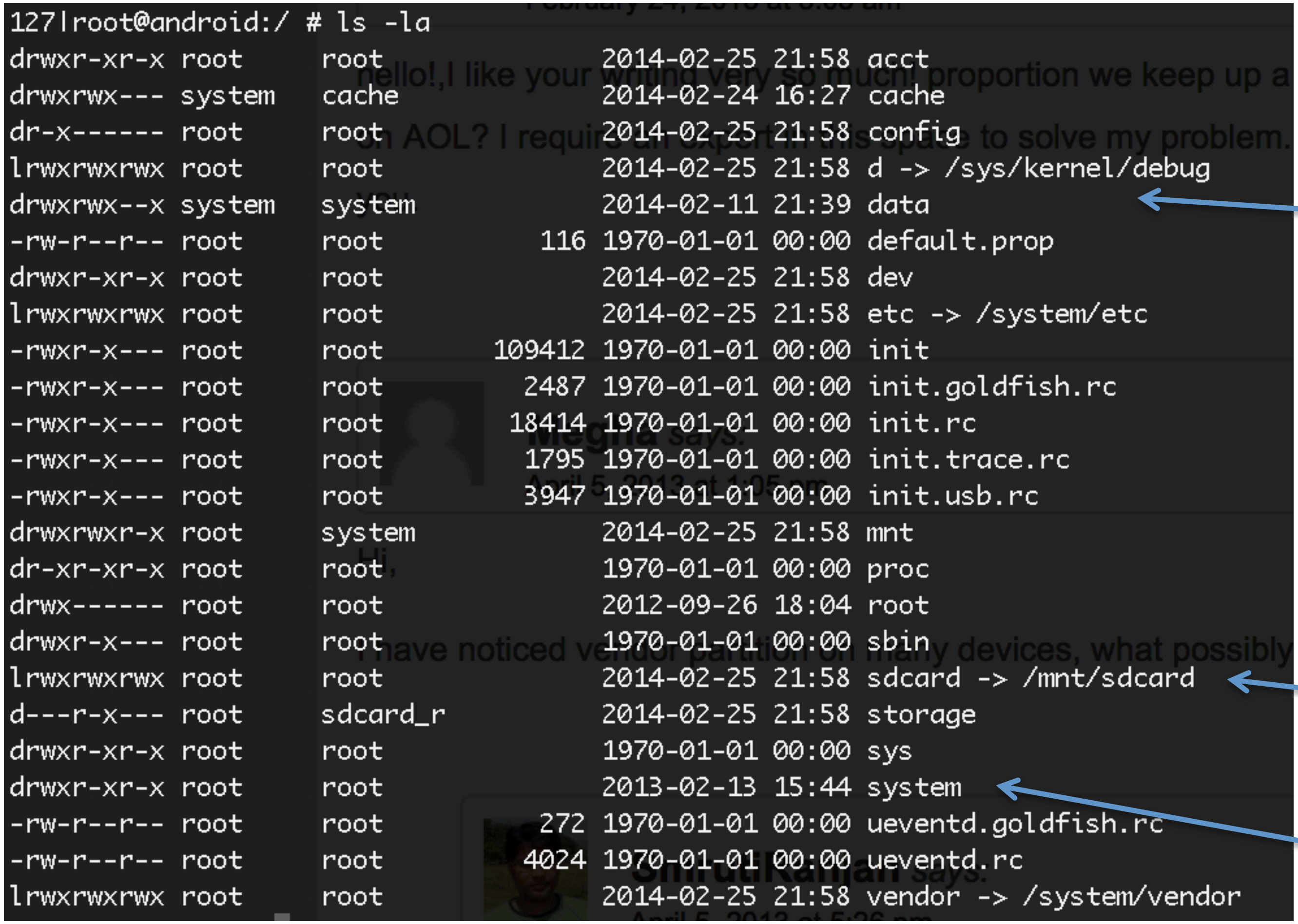
G54MDP
Mobile Device Programming

Lecture 11 – Storage

Logical Data Storage on Android

- File-based abstractions
  - Shared Preferences
    - Simple key value pairs
  - File-based storage
    - Internal Data Storage
      - Soldered RAM
      - Internal APK resources, temporary files
    - External Data Storage
      - SD Card
      - Large media files
  - SQLite Database
    - Structured data, small binary files
- Network
  - Shared contact lists, backups
  - SyncAdapter



```
127|root@android:/ # ls -la
drwxr-xr-x root     root              2014-02-25 21:58 acct
drwxrwx--- system   cache             2014-02-24 16:27 cache
dr-x------ root     root              2014-02-25 21:58 config
lrwxrwxrwx root     root              2014-02-25 21:58 d -> /sys/kernel/debug
drwxrwx--x system   system            2014-02-11 21:39 data
-rw-r--r-- root     root          116 1970-01-01 00:00 default.prop
drwxr-xr-x root     root              2014-02-25 21:58 dev
lrwxrwxrwx root     root              2014-02-25 21:58 etc -> /system/etc
-rwxr-x--- root     root       109412 1970-01-01 00:00 init
-rwxr-x--- root     root         2487 1970-01-01 00:00 init.goldfish.rc
-rwxr-x--- root     root        18414 1970-01-01 00:00 init.rc
-rwxr-x--- root     root         1795 1970-01-01 00:00 init.trace.rc
-rwxr-x--- root     root         3947 1970-01-01 00:00 init.usb.rc
drwxrwxr-x root     system            2014-02-25 21:58 mnt
dr-xr-xr-x root     root              1970-01-01 00:00 proc
drwx------ root     root              2012-09-26 18:04 root
drwxr-x--- root     root              1970-01-01 00:00 sbin
lrwxrwxrwx root     root              2014-02-25 21:58 sdcard -> /mnt/sdcard
d---r-x--- root     sdcard_r          2014-02-25 21:58 storage
drwxr-xr-x root     root              1970-01-01 00:00 sys
drwxr-xr-x root     root              2013-02-13 15:44 system
-rw-r--r-- root     root          272 1970-01-01 00:00 ueventd.goldfish.rc
-rw-r--r-- root     root         4024 1970-01-01 00:00 ueventd.rc
lrwxrwxrwx root     root              2014-02-25 21:58 vendor -> /system/vendor
```

“User” data –
application data

“External” storage

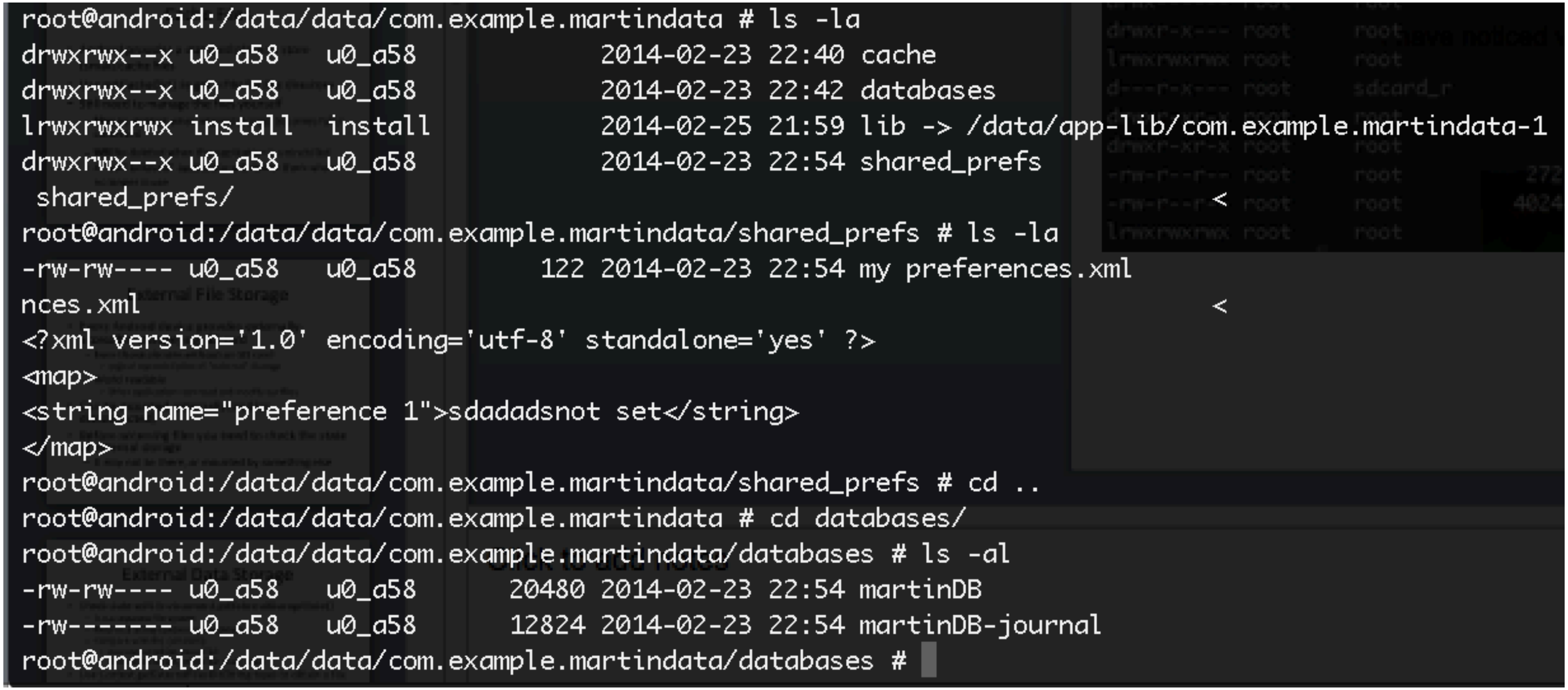
Android OS /
libraries

Internal File Storage

- Internal Data storage is private to the app
  - Other apps (and the user) cannot access it
    - Kernel enforced user permissions
  - Removed on uninstall
  - Data is stored in Files
    - openRawResource
      - Can be used to read our own packaged resources
- Two methods are used to access files on internal storage
  - Context.openFileOutput(String name, int mode)
    - Returns a FileOutputStream
  - Context.openFileInput(String name)
    - Returns a FileInputStream
  - Don't forget to catch IOExceptions

Cache Files

- Android provides a standard place to store (small) cache files
- Use getCacheDir() to get a File for the directory
- Still need to manage the files yourself
  - **May** be deleted when internal storage becomes full / contested
  - **Will** be deleted when the application is uninstalled
  - A “well behaved” application will delete them when no longer in use
  - Recommended to use less than 1MB



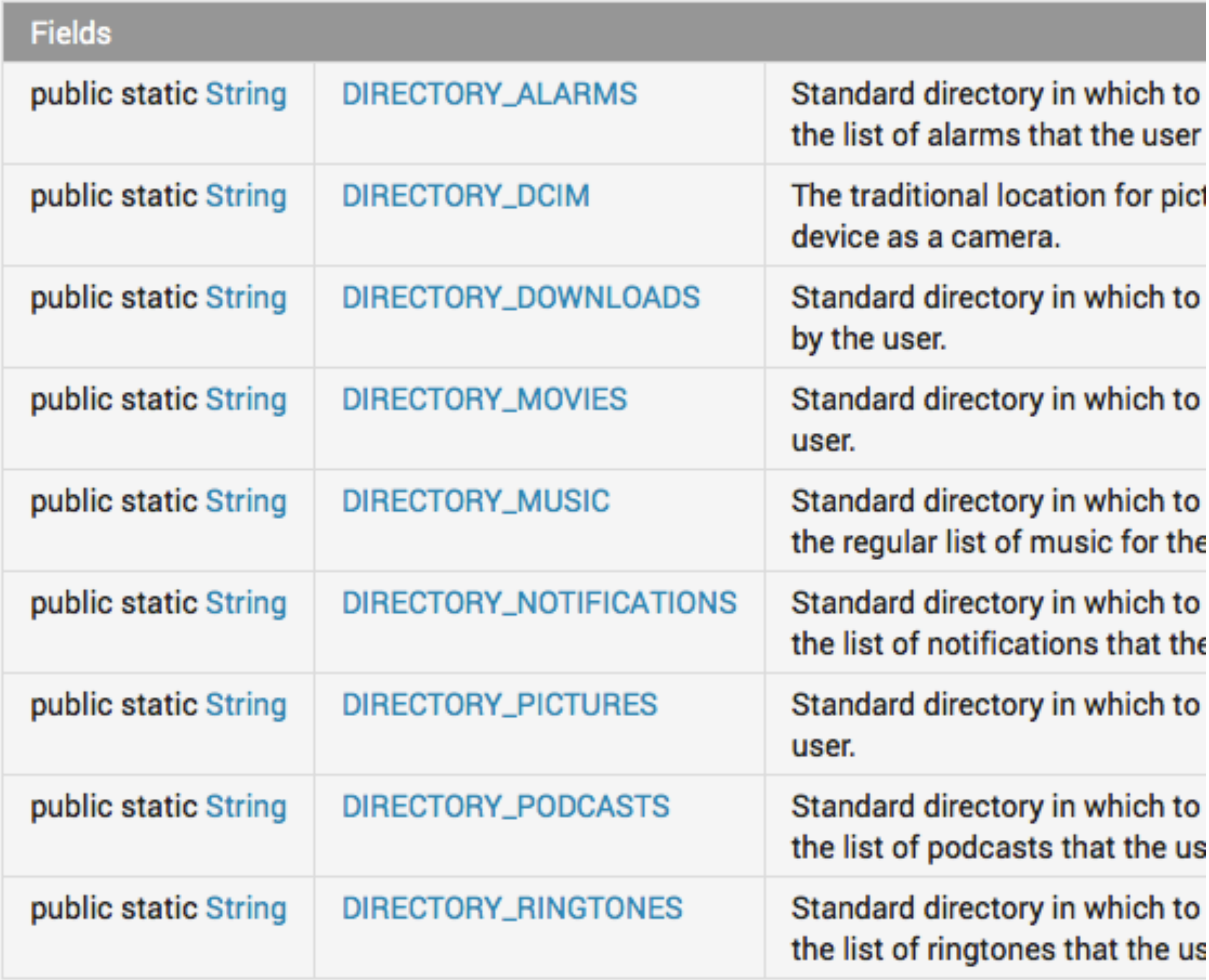
```
root@android:/data/data/com.example.martindata # ls -la
drwxrwx--x u0_a58   u0_a58            2014-02-23 22:40 cache
drwxrwx--x u0_a58   u0_a58            2014-02-23 22:42 databases
lrwxrwxrwx install  install           2014-02-25 21:59 lib -> /data/app-lib/com.example.martindata-1
drwxrwx--x u0_a58   u0_a58            2014-02-23 22:54 shared_prefs
 shared_prefs/                                                          <
root@android:/data/data/com.example.martindata/shared_prefs # ls -la
-rw-rw---- u0_a58   u0_a58        122 2014-02-23 22:54 my preferences.xml
nces.xml                                                                <
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<string name="preference 1">sdadadsnot set</string>
</map>
root@android:/data/data/com.example.martindata/shared_prefs # cd ..
root@android:/data/data/com.example.martindata # cd databases/
root@android:/data/data/com.example.martindata/databases # ls -al
-rw-rw---- u0_a58   u0_a58      20480 2014-02-23 22:54 martinDB
-rw------- u0_a58   u0_a58      12824 2014-02-23 22:54 martinDB-journal
root@android:/data/data/com.example.martindata/databases #
```

External File Storage

- Every Android device provides externally-accessible storage, e.g. SD card
  - Even those phones without an SD card
    - Logical representation of "external" storage
  - "Private" application files
  - "Public" general files
    - World readable
    - Other applications can read and modify these files
- Can be mounted externally (and/or disconnected)
- Before accessing files you need to check the state of external storage
  - It may not be there, or mounted by something else

External Data Storage

- Check state with Environment.getExternalStorageState()
  - It is a separate file system
  - Returns a String containing the details
  - Compare with the constants:
    - Environment.MEDIA_MOUNTED
    - Environment.MEDIA_MOUNTED_READ_ONLY
- Use getExternalStoragePublicDirectory(String type) to
  - If you pass a type (it can be null) then returns a sub-directory of appropriate type
  - Used to enable the Media scanner to categorize material
  - Use File object returned to createNewFile()



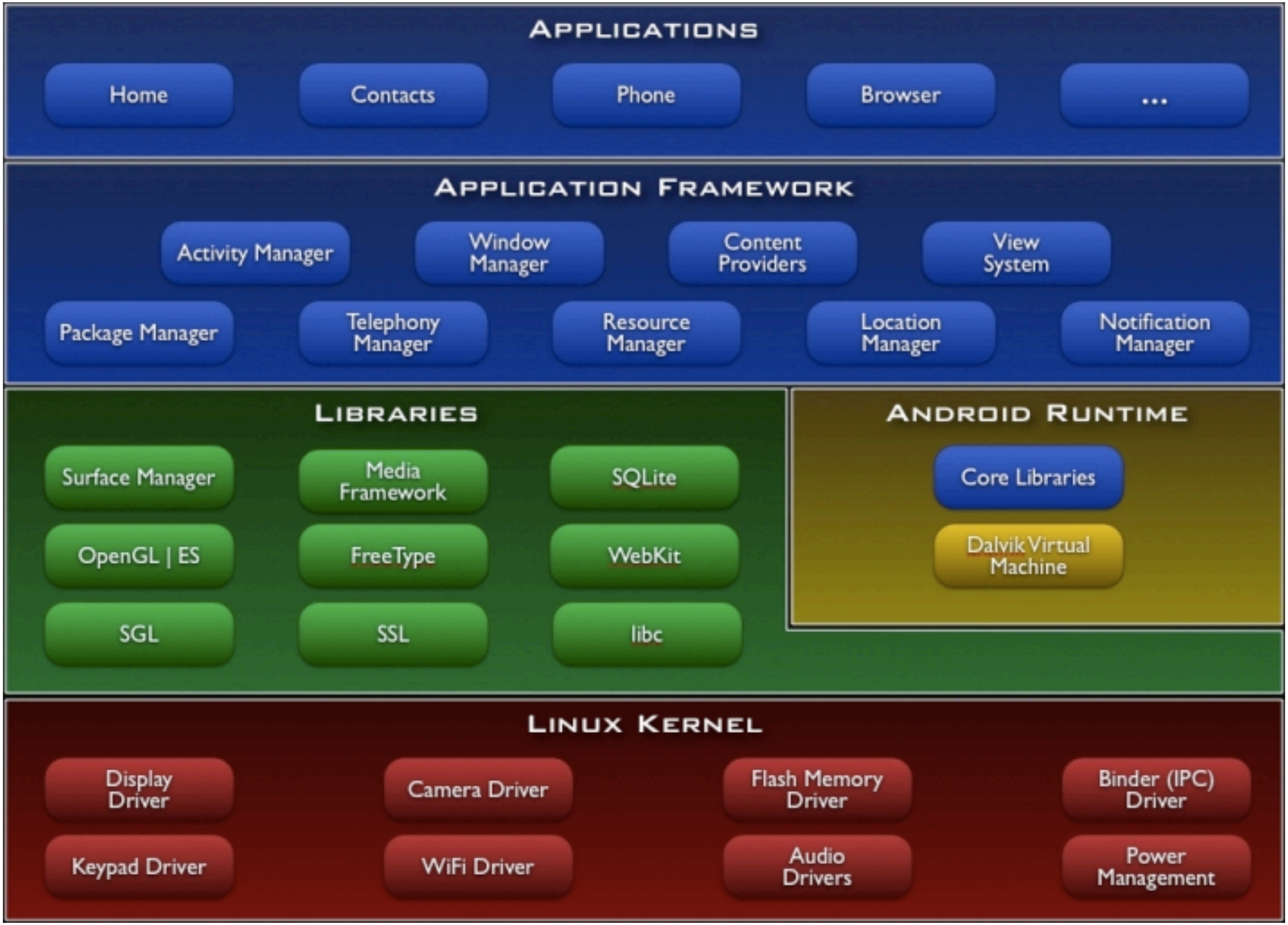
Fields		
public static String	DIRECTORY_ALARMS	Standard directory in which to the list of alarms that the user
public static String	DIRECTORY_DCIM	The traditional location for pict device as a camera.
public static String	DIRECTORY_DOWNLOADS	Standard directory in which to by the user.
public static String	DIRECTORY_MOVIES	Standard directory in which to user.
public static String	DIRECTORY_MUSIC	Standard directory in which to the regular list of music for the
public static String	DIRECTORY_NOTIFICATIONS	Standard directory in which to the list of notifications that the
public static String	DIRECTORY_PICTURES	Standard directory in which to user.
public static String	DIRECTORY_PODCASTS	Standard directory in which to the list of podcasts that the us
public static String	DIRECTORY_RINGTONES	Standard directory in which to the list of ringtones that the us

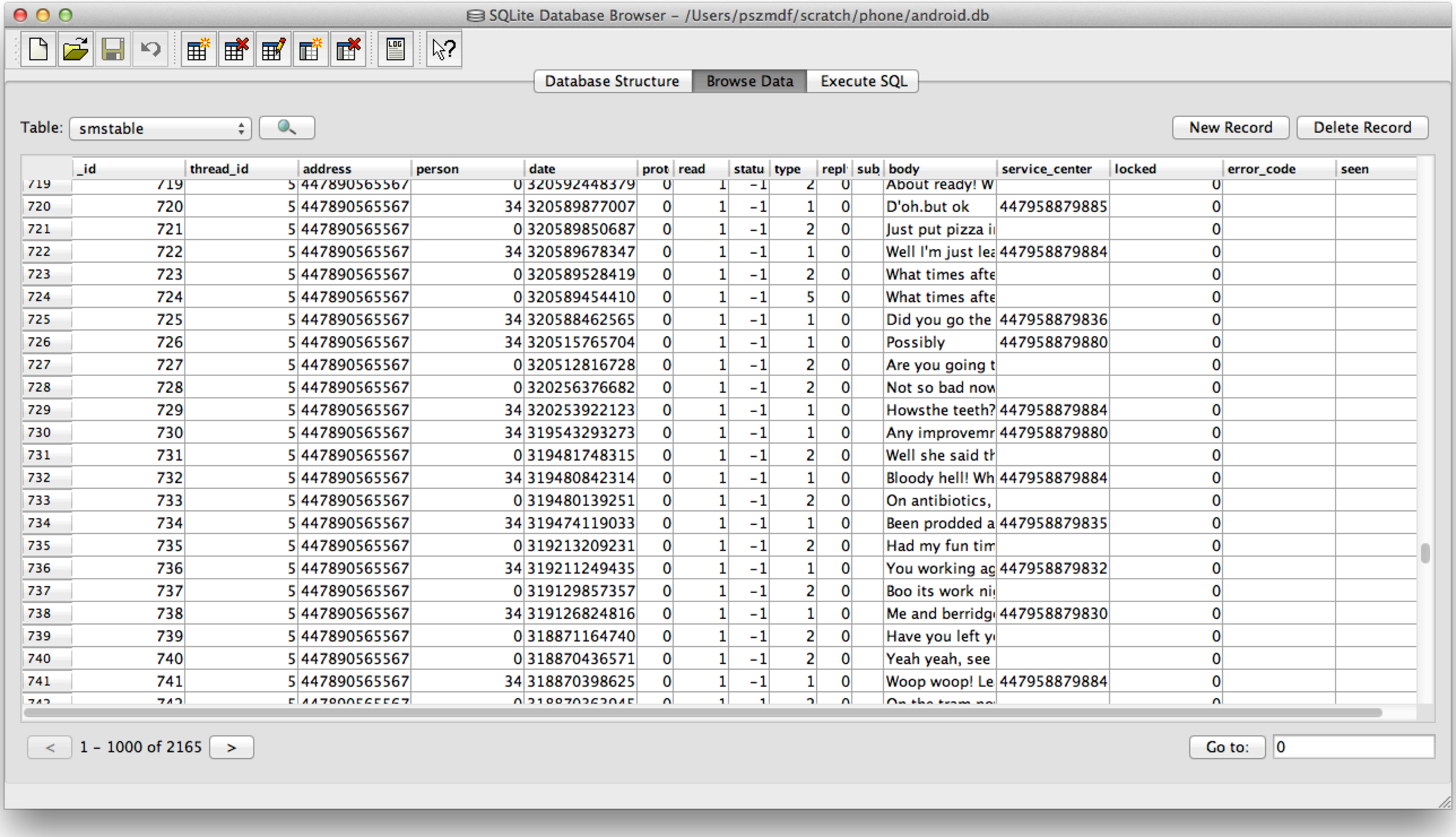
Structured Data

- Often the data we are storing is structured
- And we want to query it based on that structure
- Could store this in a file and write our own routines to access it
- Normally, we'd use a database to store it
  - – E.g. An address book, music library
  - – V.s. binary "blobs"
    - Images, mp3s
  - – Media gallery?

Android Databases

- Android comes with local database support
  - Complete with the ability to run SQL queries
  - Each app's databases are local to it
- Uses SQLite
  - Public Domain software library
  - "A software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine."
    - File based
  - "Most widely deployed software engine on the planet"


Applications
Home
Contacts
Phone
Browser
...
Application Framework
Activity Manager
Window Manager
Content Providers
View System
Package Manager
Telephony Manager
Resource Manager
Location Manager
Notification Manager
Libraries
Surface Manager
Media Framework
SQLite
OpenGL | ES
FreeType
WebKit
SGL
SSL
libc
Android Runtime
Core Libraries
Dalvik Virtual Machine
Linux Kernel
Display Driver
Camera Driver
Flash Memory Driver
Binder (IPC) Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management



SQLite Database Browser – /Users/pszmdf/scratch/phone/android.db
Database Structure
Browse Data
Execute SQL
New Record
Delete Record

Table: smstable

	_id	thread_id	address	person	date	prot	read	statu	type	repl	sub	body	service_center	locked	error_code	seen
719	719	5	447890565567	0	320592448379	0	1	-1	2	0		About ready! W		0		
720	720	5	447890565567	34	320589877007	0	1	-1	1	0		D'oh.but ok	447958879885	0		
721	721	5	447890565567	0	320589850687	0	1	-1	2	0		Just put pizza i		0		
722	722	5	447890565567	34	320589678347	0	1	-1	1	0		Well I'm just lea	447958879884	0		
723	723	5	447890565567	0	320589528419	0	1	-1	2	0		What times afte		0		
724	724	5	447890565567	0	320589454410	0	1	-1	5	0		What times afte		0		
725	725	5	447890565567	34	320588462565	0	1	-1	1	0		Did you go the	447958879836	0		
726	726	5	447890565567	34	320515765704	0	1	-1	1	0		Possibly	447958879880	0		
727	727	5	447890565567	0	320512816728	0	1	-1	2	0		Are you going t		0		
728	728	5	447890565567	0	320256376682	0	1	-1	2	0		Not so bad now		0		
729	729	5	447890565567	34	320253922123	0	1	-1	1	0		Howsthe teeth?	447958879884	0		
730	730	5	447890565567	34	319543293273	0	1	-1	1	0		Any improvem	447958879880	0		
731	731	5	447890565567	0	319481748315	0	1	-1	2	0		Well she said th		0		
732	732	5	447890565567	34	319480842314	0	1	-1	1	0		Bloody hell! Wh	447958879884	0		
733	733	5	447890565567	0	319480139251	0	1	-1	2	0		On antibiotics,		0		
734	734	5	447890565567	34	319474119033	0	1	-1	1	0		Been prodded a	447958879835	0		
735	735	5	447890565567	0	319213209231	0	1	-1	2	0		Had my fun tim		0		
736	736	5	447890565567	34	319211249435	0	1	-1	1	0		You working ag	447958879832	0		
737	737	5	447890565567	0	319129857357	0	1	-1	2	0		Boo its work ni		0		
738	738	5	447890565567	34	319126824816	0	1	-1	1	0		Me and berridg	447958879830	0		
739	739	5	447890565567	0	318871164740	0	1	-1	2	0		Have you left y		0		
740	740	5	447890565567	0	318870436571	0	1	-1	2	0		Yeah yeah, see		0		
741	741	5	447890565567	34	318870398625	0	1	-1	1	0		Woop woop! Le	447958879884	0		
742	742	5	447890565567	0	318870363045	0	1	-1	2	0		On the tram no		0		

< 1 – 1000 of 2165 >

Go to: 0

Android and SQLite

- Wrapped up in two main classes
  - Database represented by SQLiteDatabase
    - Lets us run SQL queries on the database
  - Also provides SQLiteOpenHelper to help create the database

Using Databases

- SQLiteOpenHelper manages database creation and upgrades between versions
  - Create a subclass of it
  - Override onCreate to provide the code to create the database
  - Using SQL CREATE TABLE
  - Handled automatically
- Create an instance of our SQLiteOpenHelper subclass
- Obtain reference to SQLiteDatabase using:
  - getReadableDatabase()
  - getWriteableDatabase()
- Both return the same object, unless memory is low and can only open the DB readonly

Querying a Database

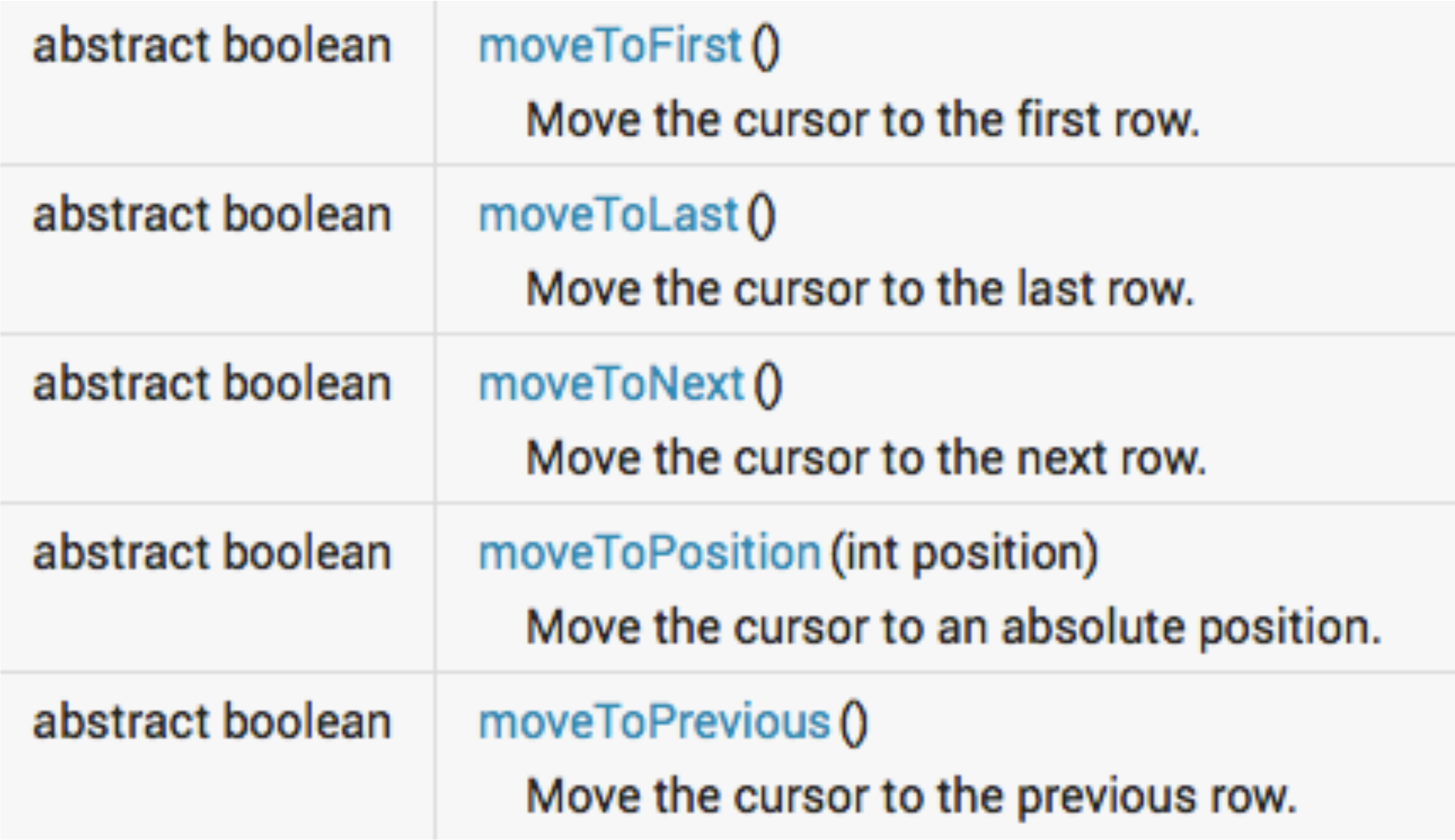
- SQLiteDatabase has many methods
- void execSQL()
  - used to run SQL queries that don’t return anything
- More useful are query() and rawQuery()
  - These return a Cursor object that can be used to access the data
  - “Move” the Cursor around the results
  - Provides random access to the results

Querying a Database

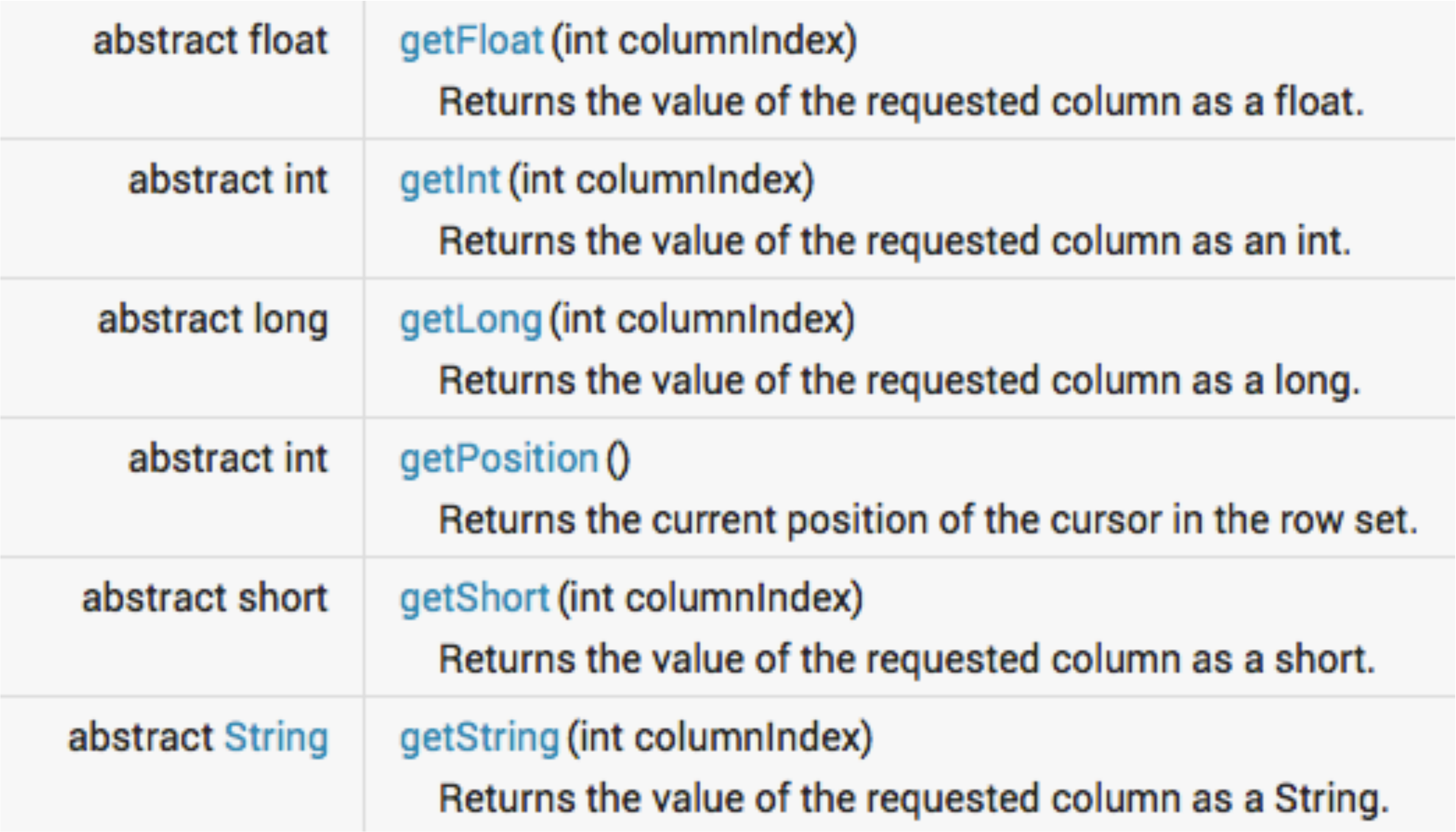
- `Cursor rawQuery(String sql, String[] selectionArgs)`
  - processes a raw SQL query
  - `rawQuery("SELECT id, name FROM people WHERE name = ? AND id = ?", new String[] {”Martin", ”78"});`
- SQL has to be parsed so there is also query() where the SQL is exploded into separate strings
  - Simpler to construct a query programmatically
  - `Cursor query(String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy)`

Cursors

- Provides random access to results of a query
- Fairly self explanatory object
  - Enables you to step over all the rows returned by a query
  - Has a close() method to close the query when you are finished
    - don't wait for it to be garbage collected
  - "Connect" a cursor to a CursorAdapter and ListView



abstract boolean	moveToFirst () Move the cursor to the first row.
abstract boolean	moveToLast () Move the cursor to the last row.
abstract boolean	moveToNext () Move the cursor to the next row.
abstract boolean	moveToPosition (int position) Move the cursor to an absolute position.
abstract boolean	moveToPrevious () Move the cursor to the previous row.



abstract float	getFloat (int columnIndex) Returns the value of the requested column as a float.
abstract int	getInt (int columnIndex) Returns the value of the requested column as an int.
abstract long	getLong (int columnIndex) Returns the value of the requested column as a long.
abstract int	getPosition () Returns the current position of the cursor in the row set.
abstract short	getShort (int columnIndex) Returns the value of the requested column as a short.
abstract String	getString (int columnIndex) Returns the value of the requested column as a String.

Database Abstraction

- Good software architecture
  - Separation of data model from presentation / views
- Abstraction of database architecture
  - Easier to update storage code
  - Expose column indices as static class variables
    - c.getInt(0) -> c.getInt(DBHelper.NAME)
  - Helper methods keep database internals from "leaking" into other classes
    - Return a Collection of results rather than a Cursor
      - Use Cursor internally in DBHelper class
  - SQL injection
    - Sanitise user input

Databases in short

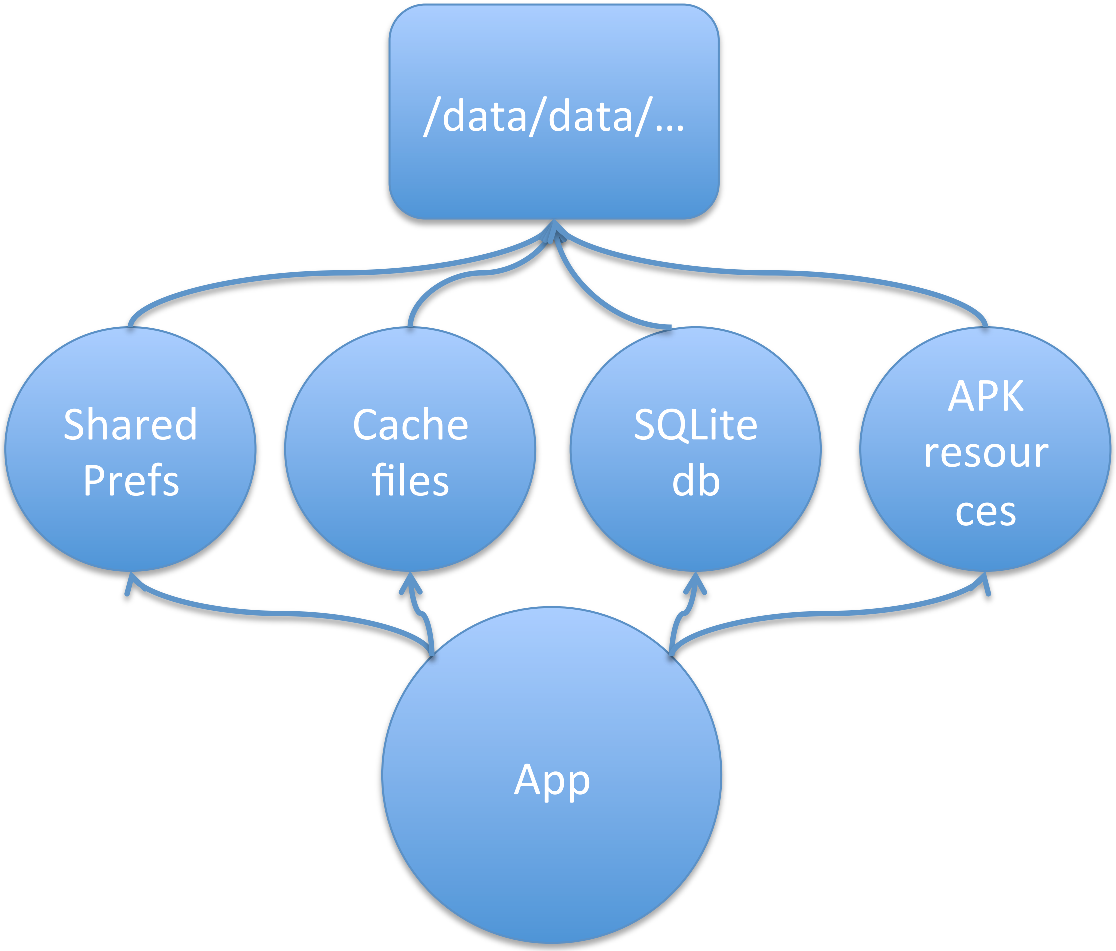
- Subclass SQLiteOpenHelper to create a database
- Use execSQL to create tables and insert data
- Use query to query the database and return multiple rows
- Manipulate a Cursor object to extract data from a query

Let’s have a look…

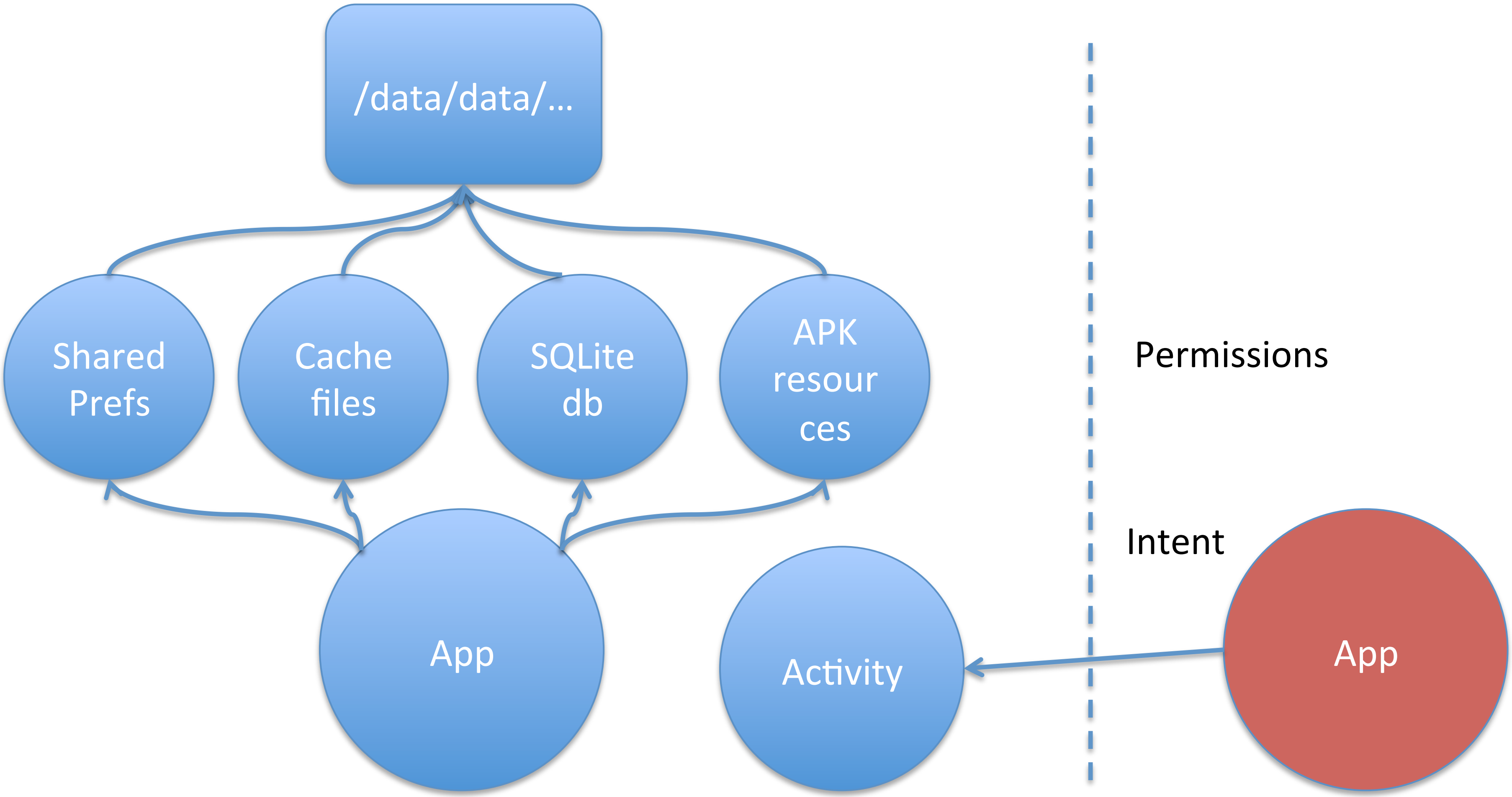
References

- http://developer.android.com/guide/topics/data/data-storage.html
- http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html
- http://developer.android.com/reference/android/database/Cursor.html

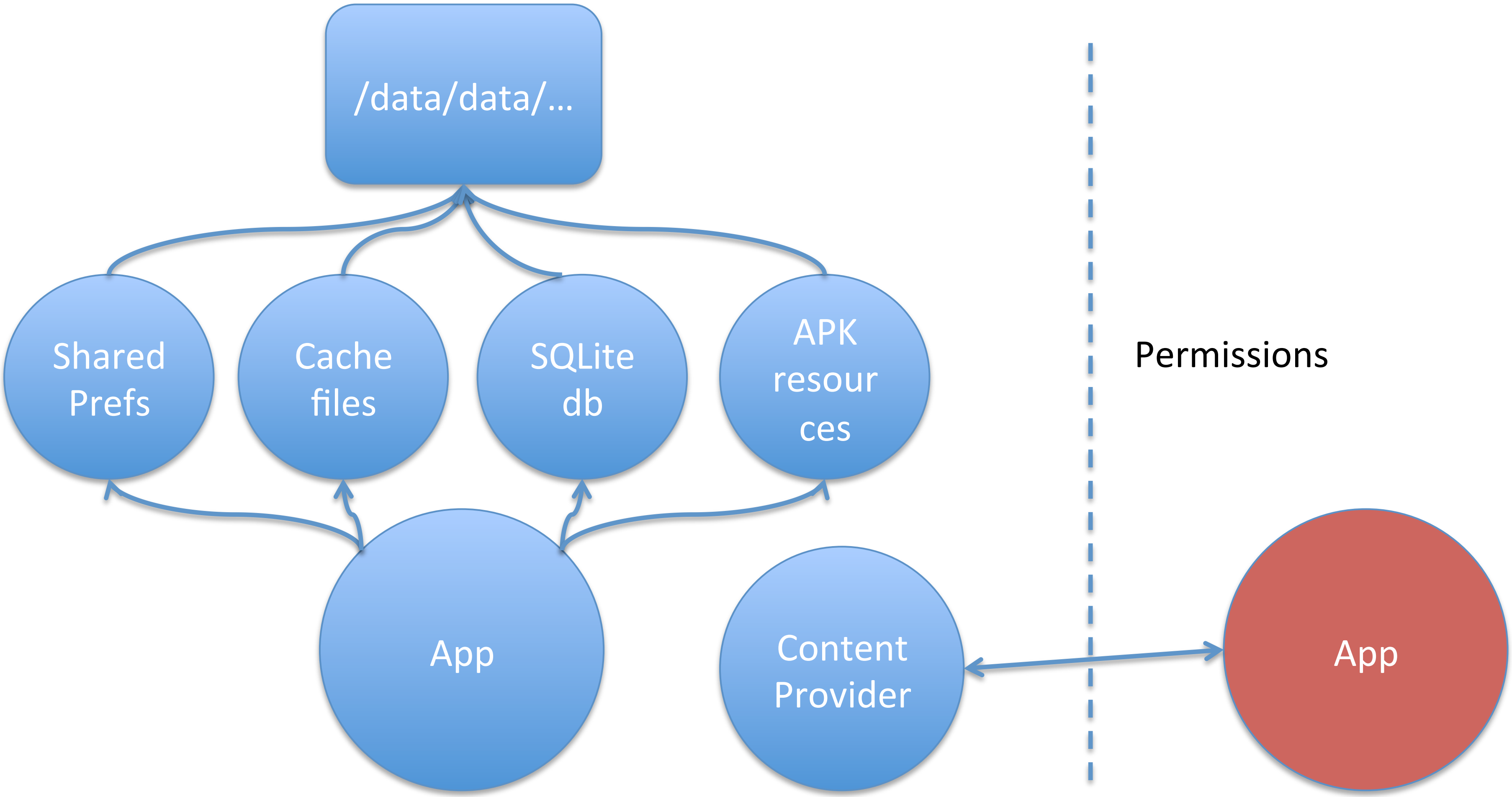
Sharing Data



Sharing Data – is this good enough?



Sharing Data – if not



ContentProvider

- Access to data is restricted to the app that owns it
  - Remember where the database file is?
  - If we want other apps to access our data, or we want to access other apps' data
  - …we need to provide or make use of a ContentProvider
    - Component number **3**
    - Exposes data / content to other applications in a structured manner
    - Fundamentally IPC via Binder with a strict interface

System ContentProviders

- ContentProviders manage data for:
  - Browser
    - Bookmarks, history
  - Call log
    - Telephone usage
  - Contacts
    - Contact data
    - WhatsApp?
  - Media
    - Media database
  - UserDictionary
    - Database for predictive spelling
  - …
- Again, recall common mobile capabilities