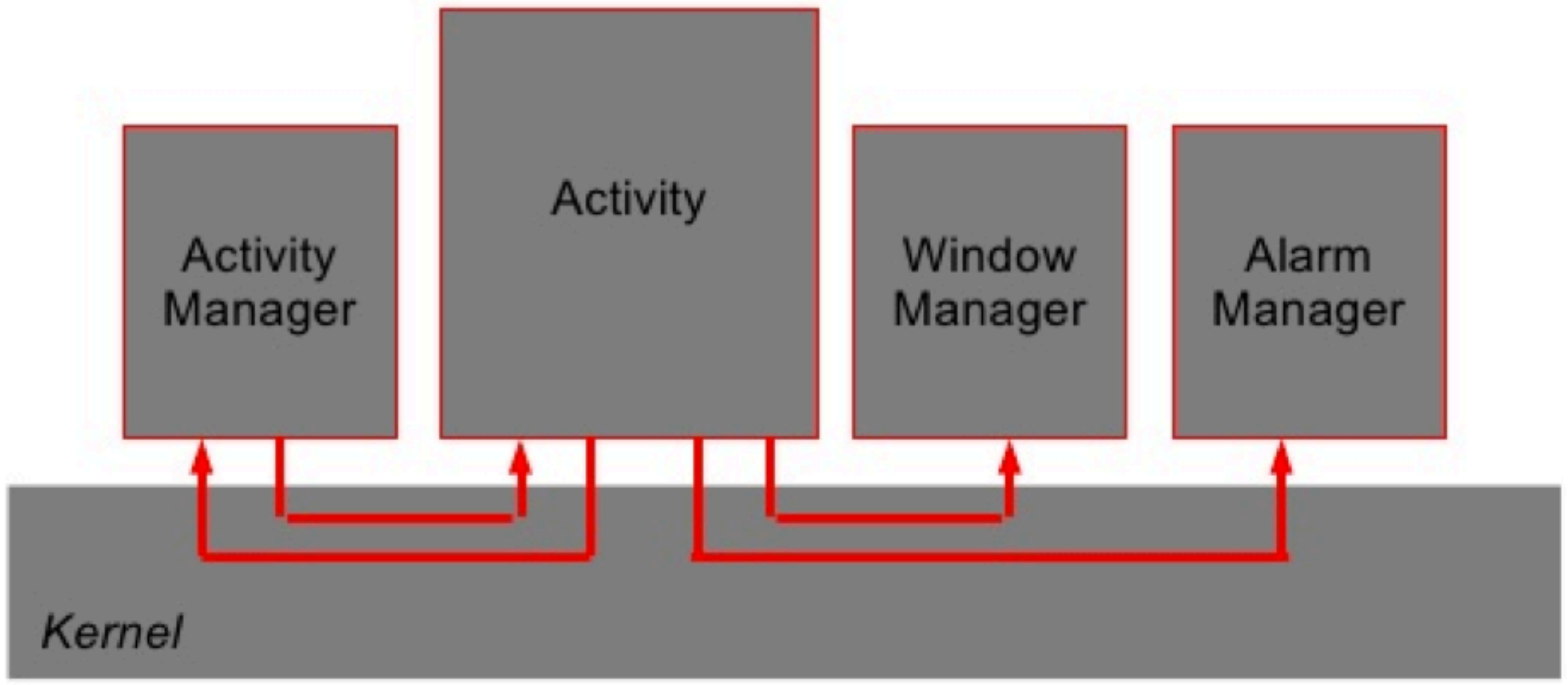
G54MDP
Mobile Device Programming

Lecture 10 – IPC, Storage

IPC – Inter-Process Communication

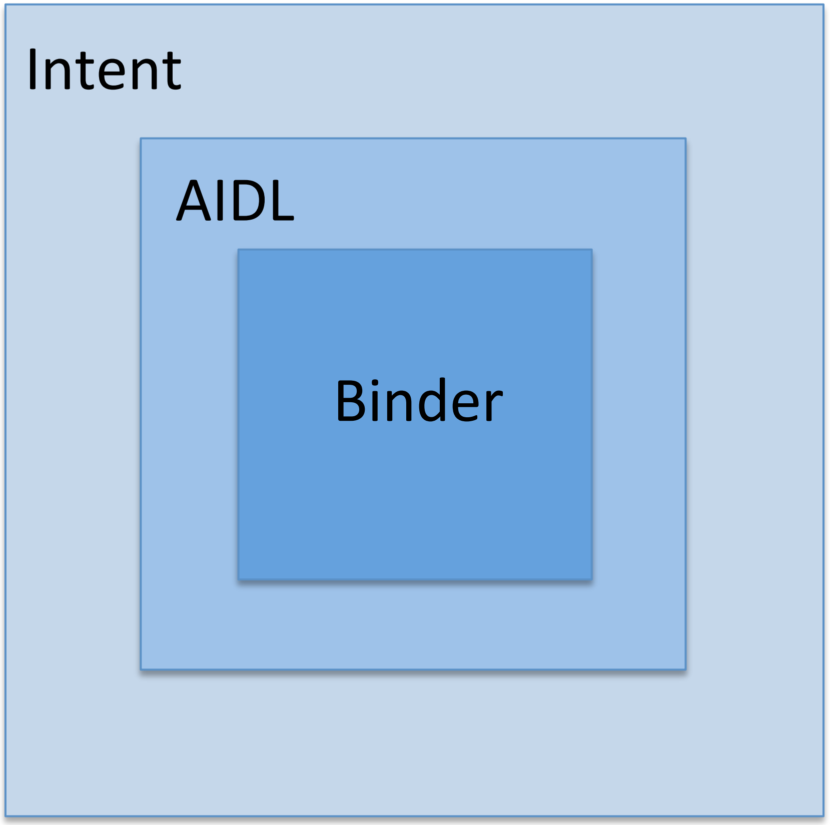


IPC

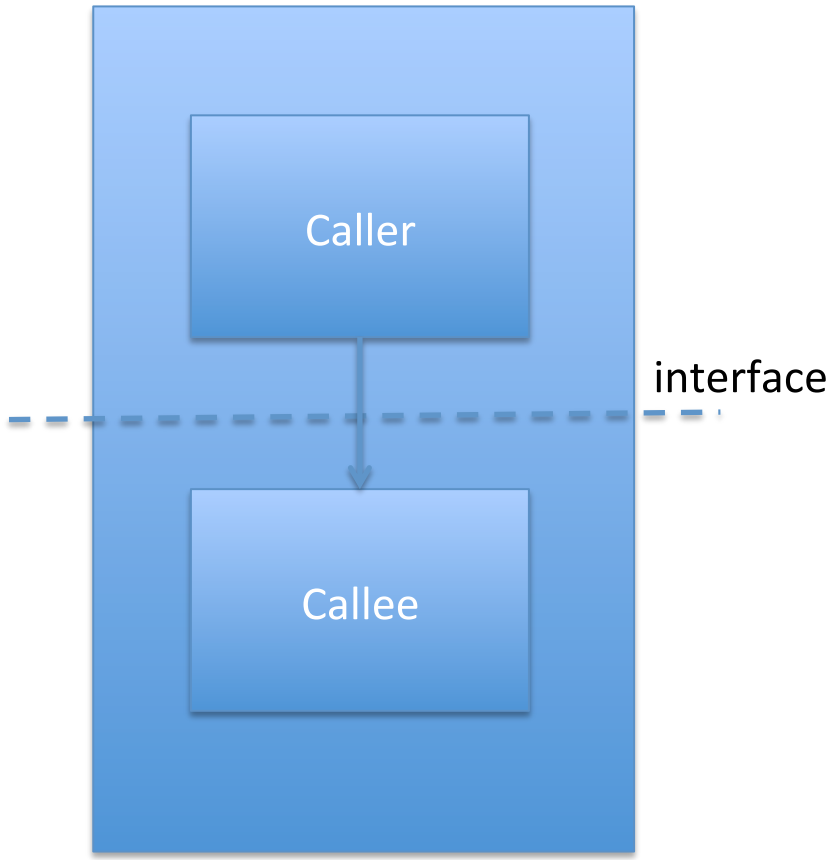
- Each process has its own address space
  - Provides data isolation
  - Prevents direct interaction between different processes
    - However, often required for modularisation
- What **actually** happens when we start a Service, or send an Intent?
- Binder
  - Underpins most Android communication
    - i.e. when we use the NotificationManager
  - Provides lightweight RPC (remote procedure communication)
    - C.f. Linux/Unix signals / pipes / sockets etc
  - Kernel driver
  - High performance via shared memory
  - Per-process thread pool for handling requests
  - Synchronous calls between processes

IPC Abstraction

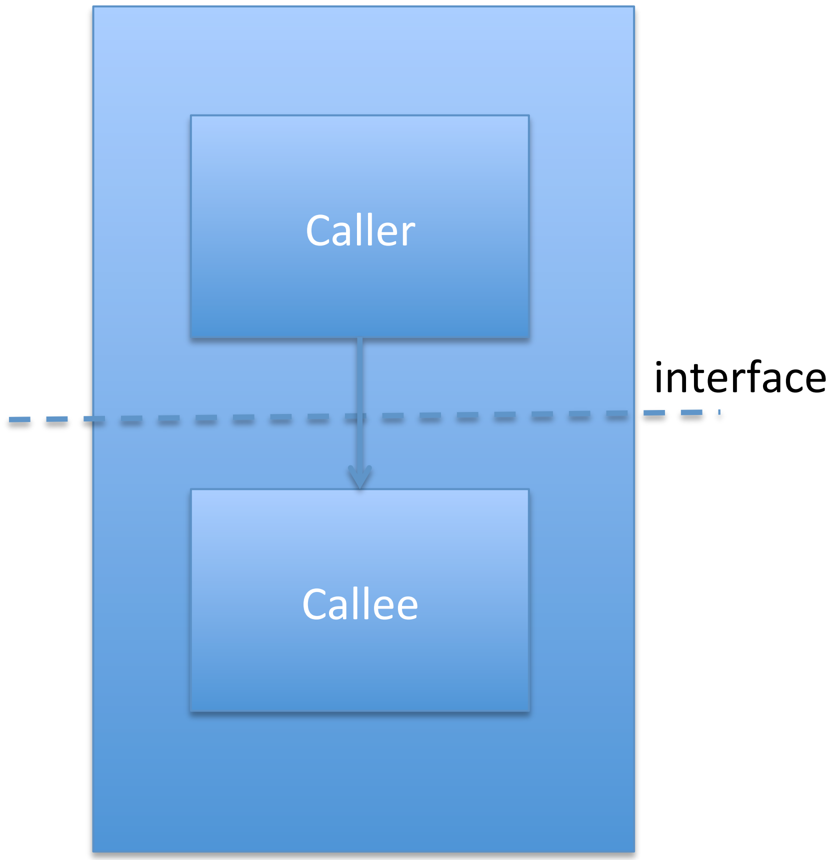
- Intent
  - Highest level abstraction
- Inter process method invocation
  - AIDL
- binder: kernel driver
- ashmem: shared memory

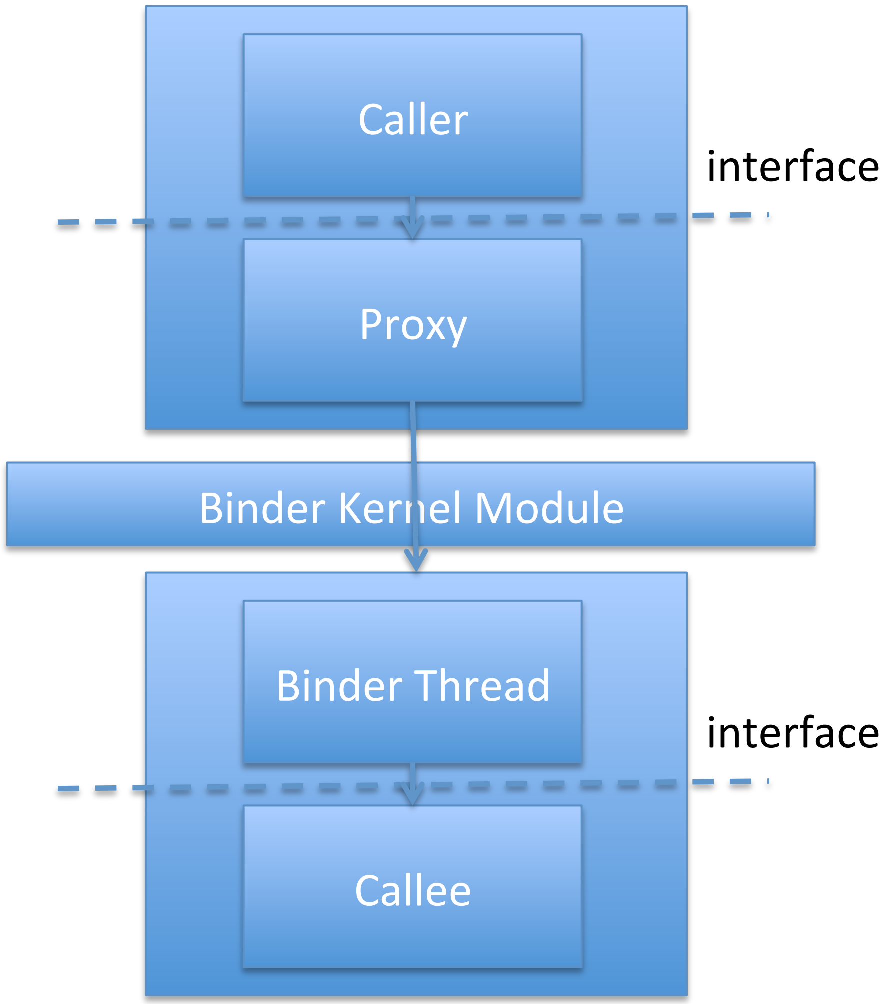


Inter-process method invocation

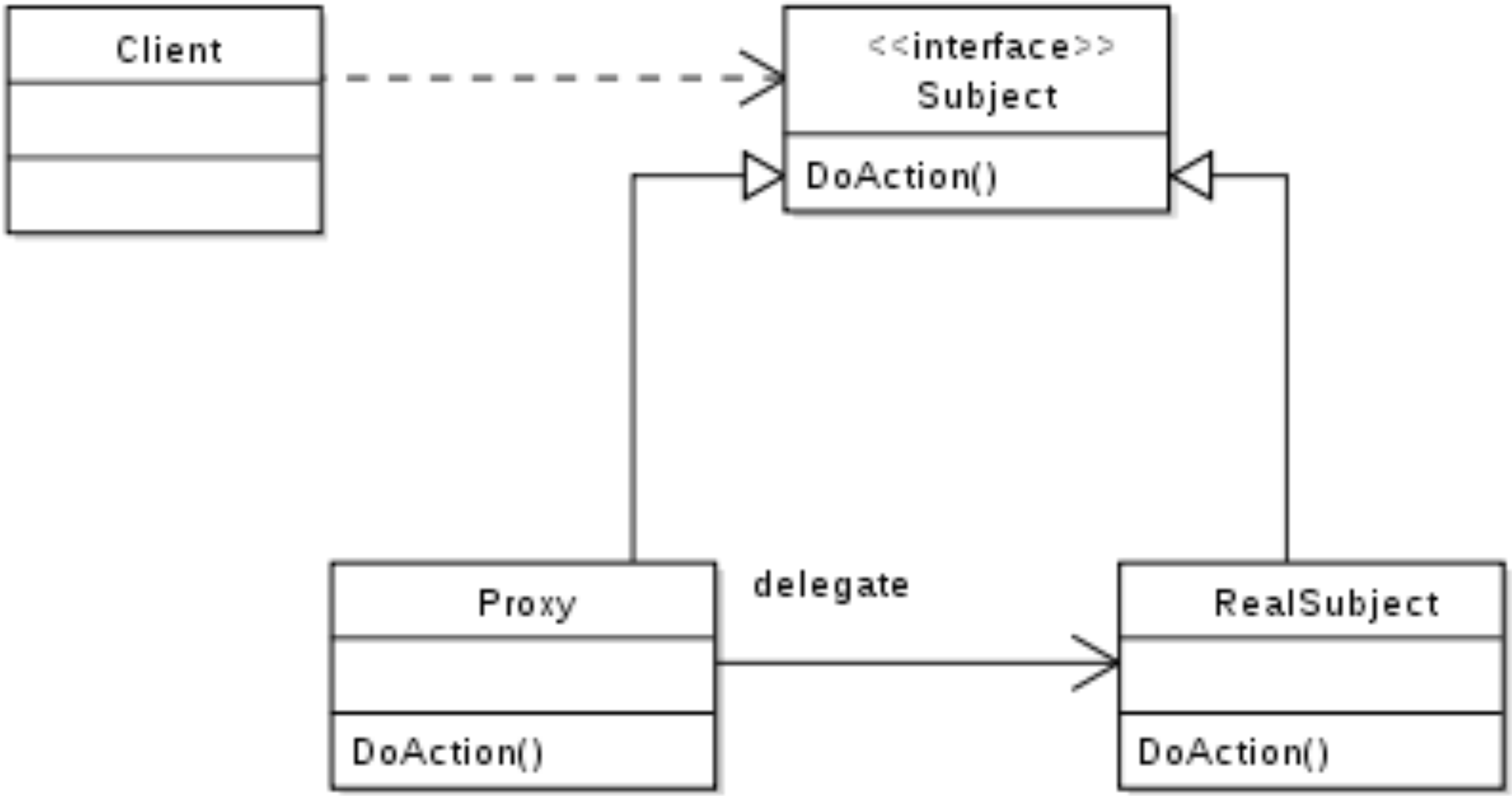


Inter-process method invocation



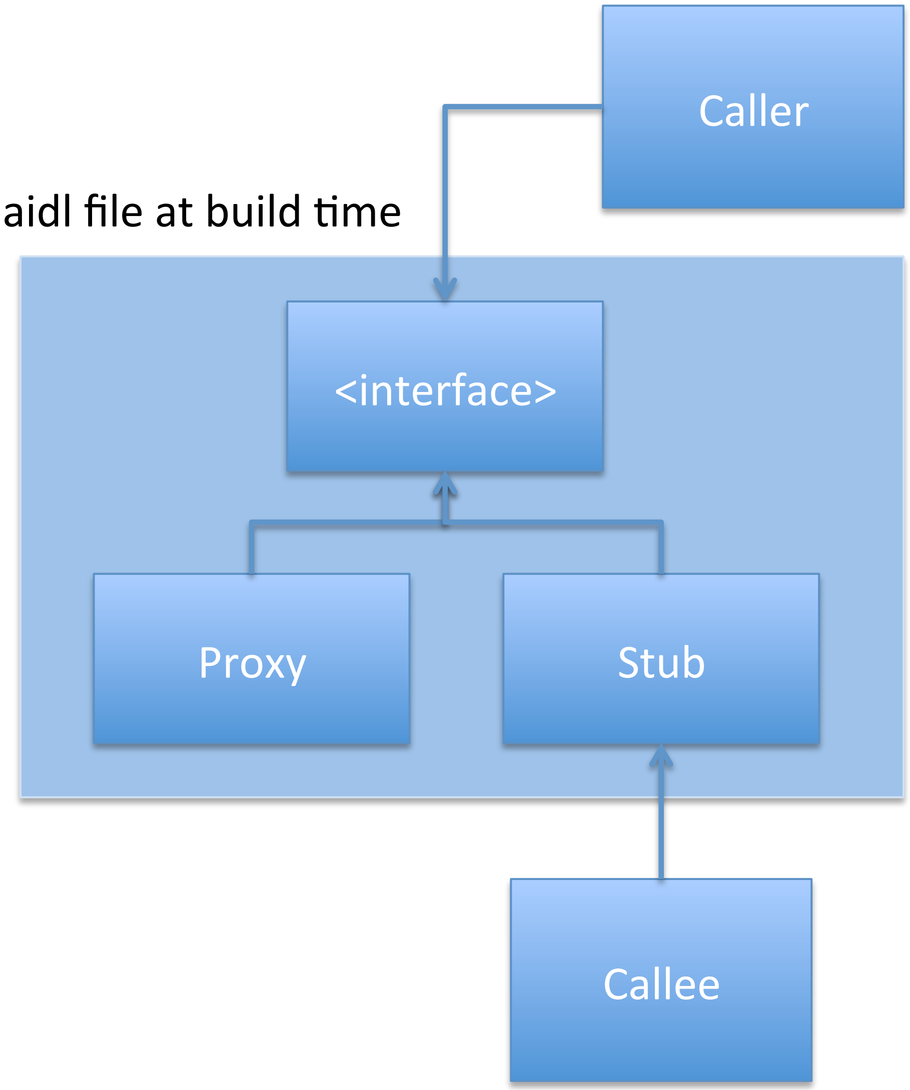


Proxy Design Pattern

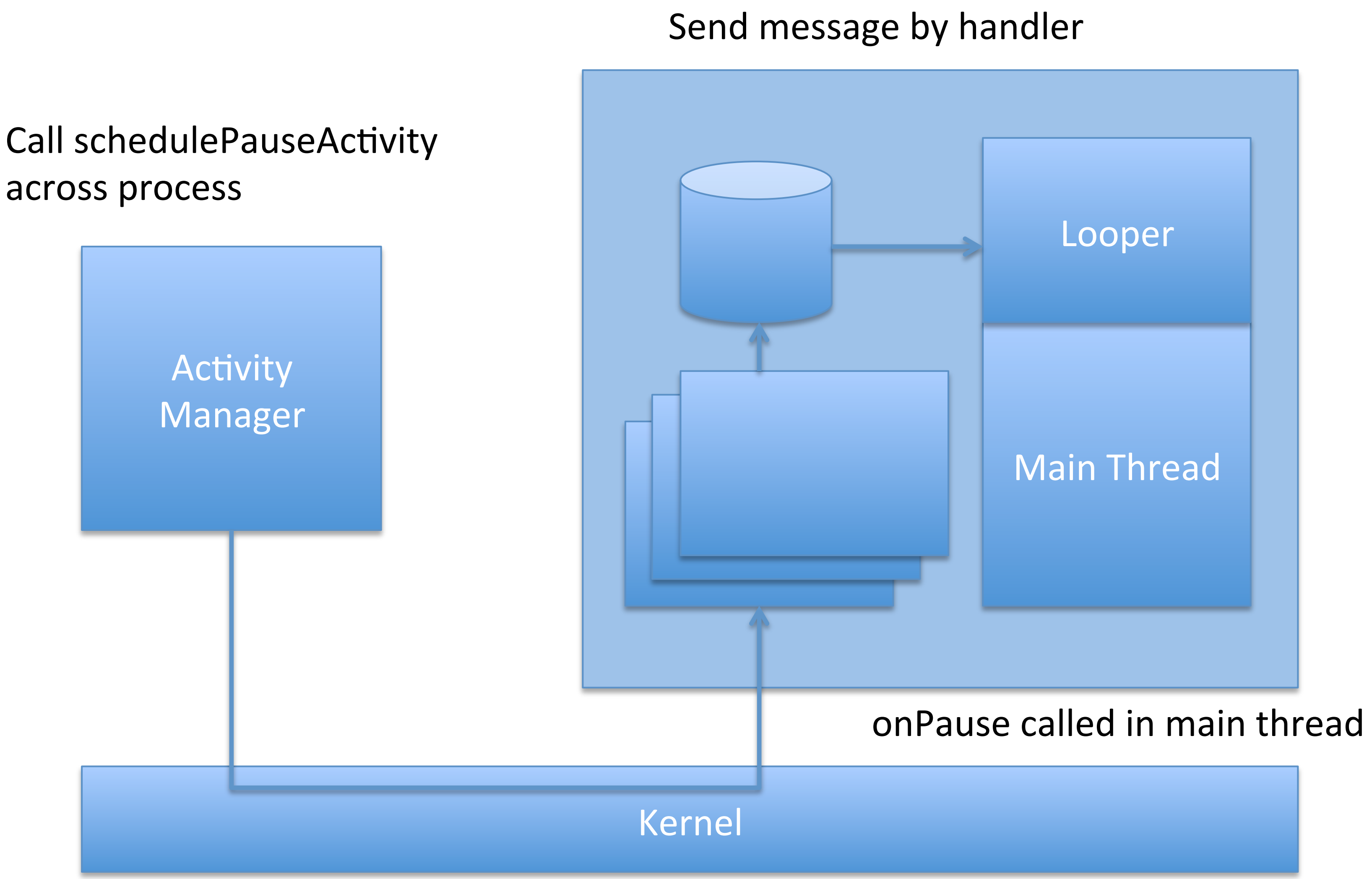


AIDL

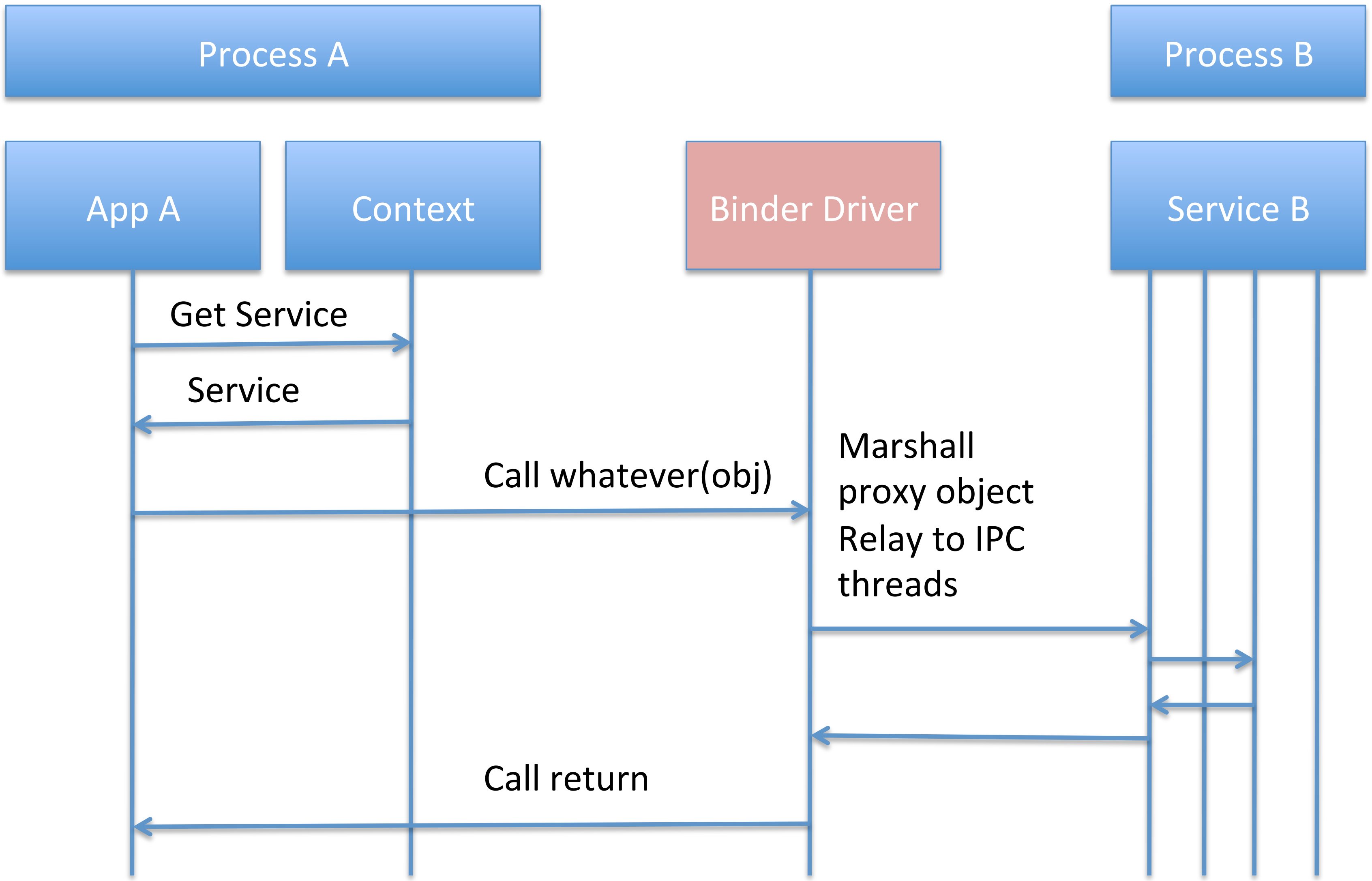
Auto generated from .aidl file at build time



onPause()


Send message by handler
Call schedulePauseActivity across process
Looper
Activity Manager
Main Thread
onPause called in main thread
Kernel

Binder in action

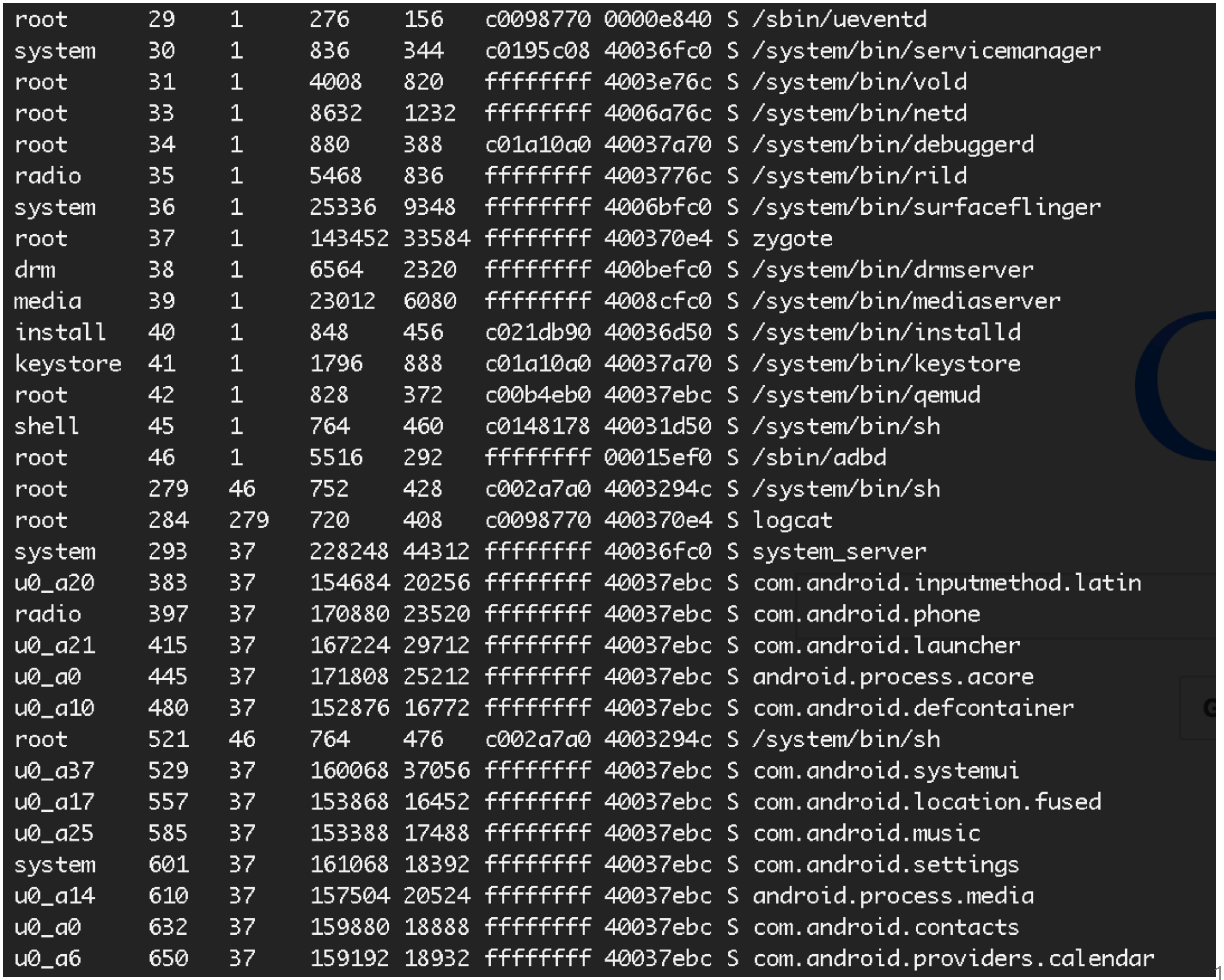


Binder Functionality

- Architecture
  - Binder kernel driver
  - Instance of Binder objects within user-space
    - Implements the IBinder interface
- Managing communication between processes
  - Simple inter-process messaging
    - Parcelable objects
  - Inter-process message calls
    - Call methods on remote objects as if they were local
  - Notifying processes of service events
- Identifying processes and services
  - Binder Token
    - Numerically uniquely identify a Binder instance
  - Basis of Android's **permissions** model
    - What are processes allowed to do?

ServiceManager

- A special Binder instance with a known Binder address
  - Hosts many system services within its process
  - Knows about other remote services
- Client does not know the token of remote Binder
  - Only the Binder interface knows its own address
- Binder submits a service name and its Binder token to the ServiceManager via IPC
  - Client retrieves remote service Binder address with service name
  - Client communicates with remote service

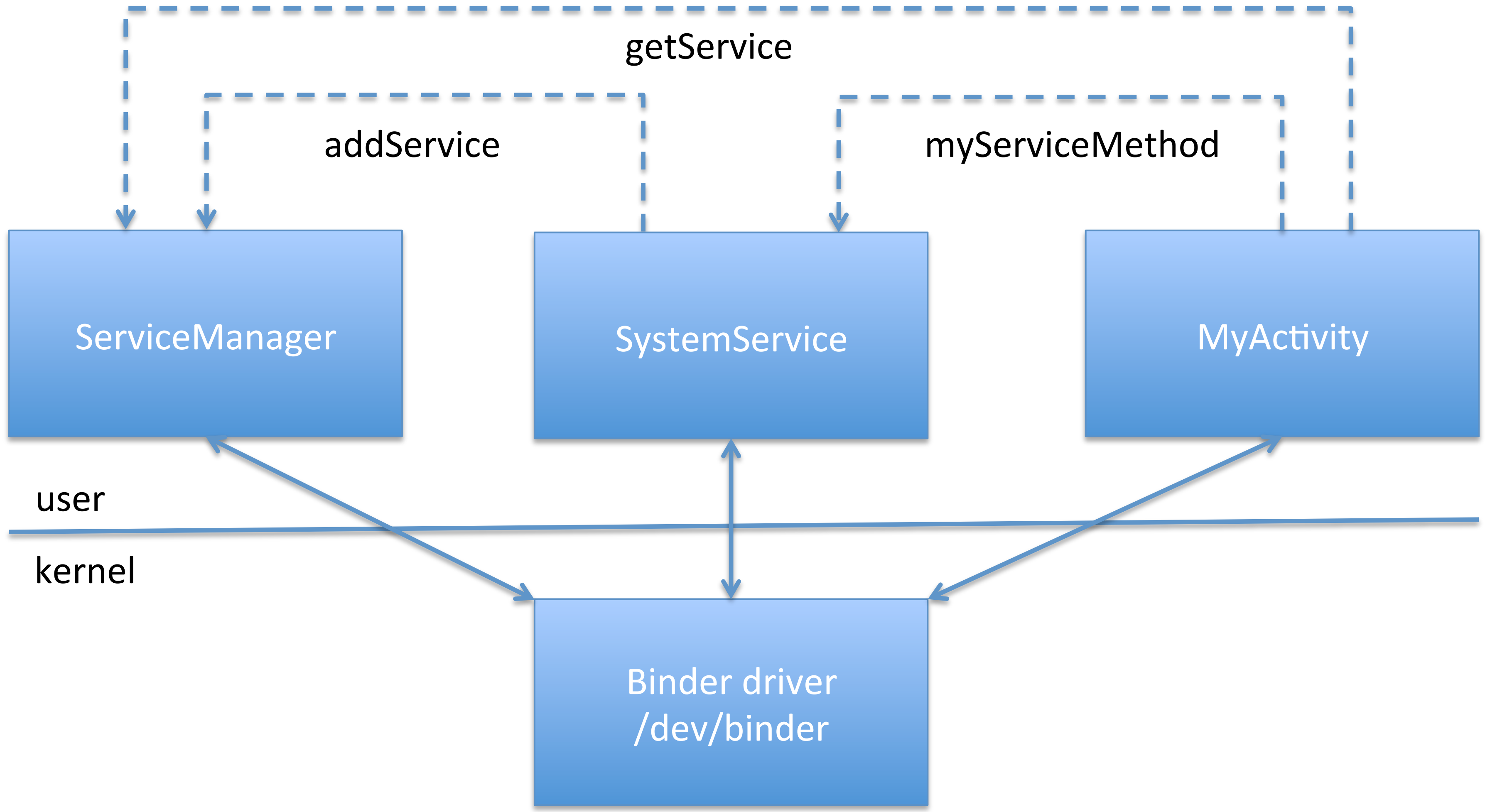


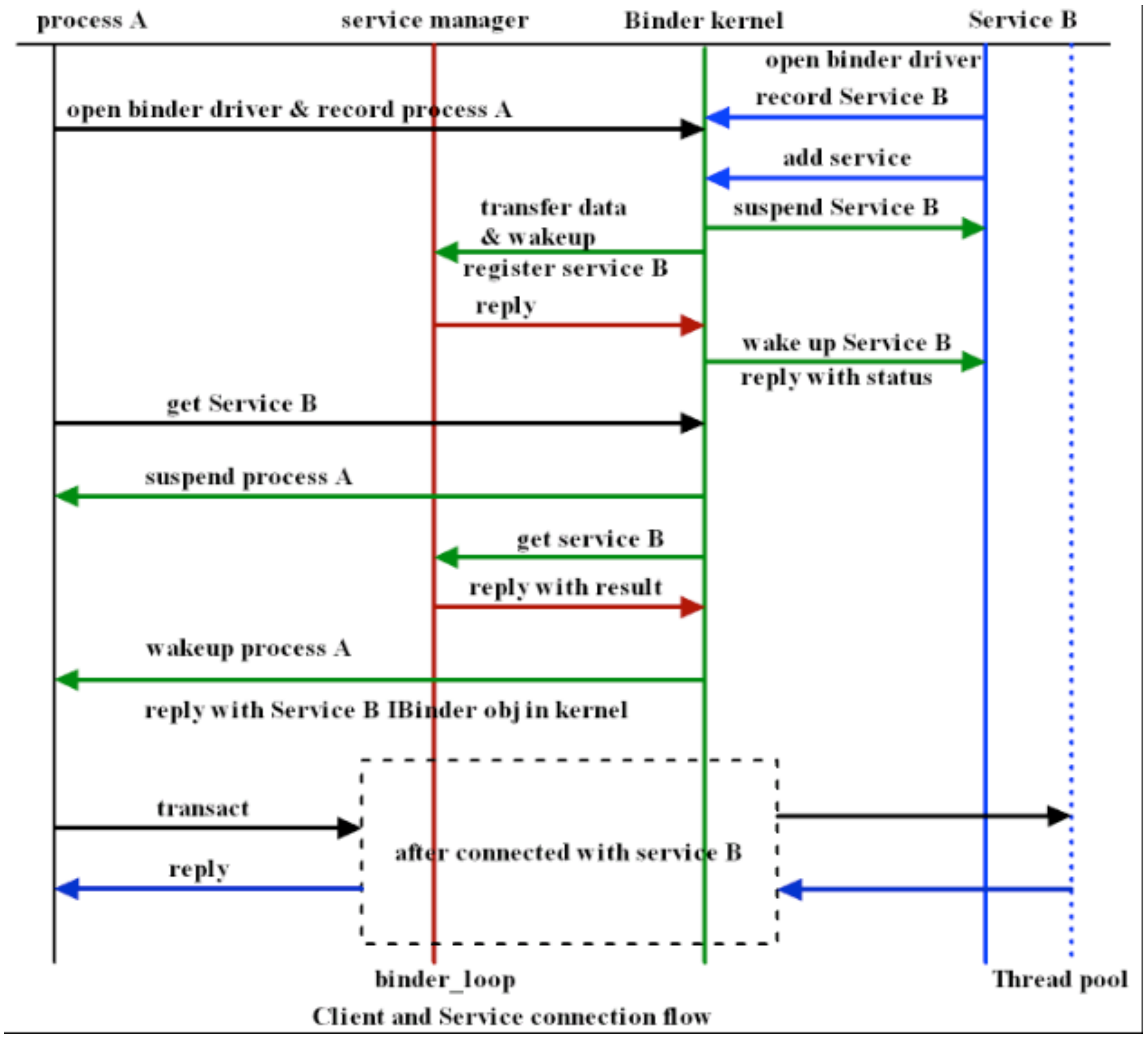
```
root      29    1     276    156   c0098770 0000e840 S /sbin/ueventd
system    30    1     836    344   c0195c08 40036fc0 S /system/bin/servicemanager
root      31    1     4008   820   ffffffff 4003e76c S /system/bin/vold
root      33    1     8632   1232  ffffffff 4006a76c S /system/bin/netd
root      34    1     880    388   c01a10a0 40037a70 S /system/bin/debuggerd
radio     35    1     5468   836   ffffffff 4003776c S /system/bin/rild
system    36    1     25336  9348  ffffffff 4006bfc0 S /system/bin/surfaceflinger
root      37    1     143452 33584 ffffffff 400370e4 S zygote
drm       38    1     6564   2320  ffffffff 400befc0 S /system/bin/drmserver
media     39    1     23012  6080  ffffffff 4008cfc0 S /system/bin/mediaserver
install   40    1     848    456   c021db90 40036d50 S /system/bin/installd
keystore  41    1     1796   888   c01a10a0 40037a70 S /system/bin/keystore
root      42    1     828    372   c00b4eb0 40037ebc S /system/bin/qemud
shell     45    1     764    460   c0148178 40031d50 S /system/bin/sh
root      46    1     5516   292   ffffffff 00015ef0 S /sbin/adbd
root      279   46    752    428   c002a7a0 4003294c S /system/bin/sh
root      284   279   720    408   c0098770 400370e4 S logcat
system    293   37    228248 44312 ffffffff 40036fc0 S system_server
u0_a20    383   37    154684 20256 ffffffff 40037ebc S com.android.inputmethod.latin
radio     397   37    170880 23520 ffffffff 40037ebc S com.android.phone
u0_a21    415   37    167224 29712 ffffffff 40037ebc S com.android.launcher
u0_a0     445   37    171808 25212 ffffffff 40037ebc S android.process.acore
u0_a10    480   37    152876 16772 ffffffff 40037ebc S com.android.defcontainer
root      521   46    764    476   c002a7a0 4003294c S /system/bin/sh
u0_a37    529   37    160068 37056 ffffffff 40037ebc S com.android.systemui
u0_a17    557   37    153868 16452 ffffffff 40037ebc S com.android.location.fused
u0_a25    585   37    153388 17488 ffffffff 40037ebc S com.android.music
system    601   37    161068 18392 ffffffff 40037ebc S com.android.settings
u0_a14    610   37    157504 20524 ffffffff 40037ebc S android.process.media
u0_a0     632   37    159880 18888 ffffffff 40037ebc S com.android.contacts
u0_a6     650   37    159192 18932 ffffffff 40037ebc S com.android.providers.calendar
```

Services

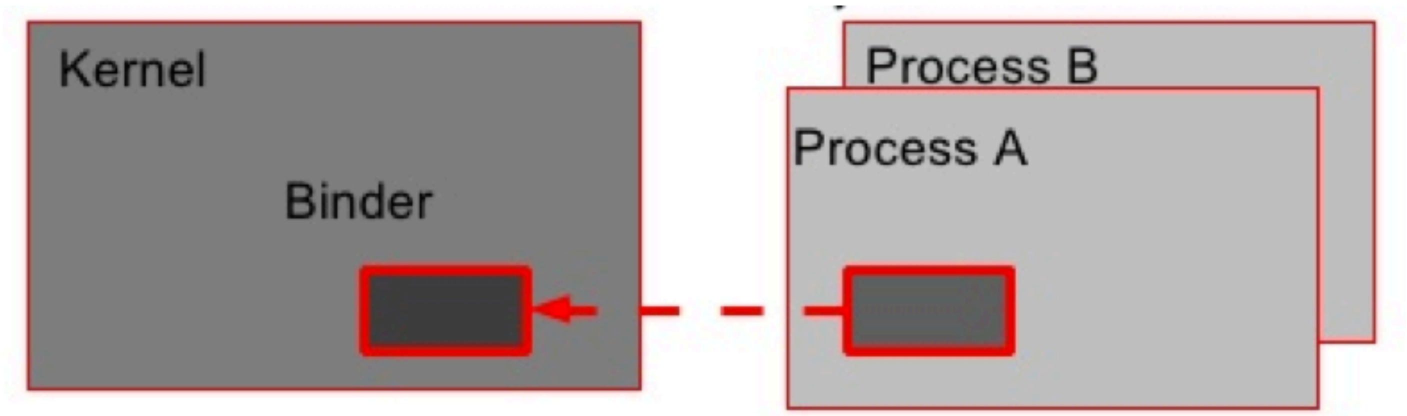
- Entropy Service
- Power Manager
- Activity Manager
- Telephony Registry
- Package Manager
- Account Manager
- Content Manger
- System Content Providers
- Battery Service
- Lights Service
- Vibrator Service
- Alarm Manager
- Init Watchdog
- Window Manager
- Bluetooth Service
- Device Policy
- Status Bar
- Clipboard Service
- Input Method Service
- NetStat Service
- NetworkManagement Service
- Connectivity Service
- Throttle Service
- Accessibility Manager
- Mount Service
- Notification Manager
- Device Storage Monitor
- Location Manager
- Search Service
- DropBox Service
- Wallpaper Service
- Audio Service
- Headset Observer
- Dock Observer
- USB Observer
- UI Mode Manager Service
- Backup Service
- AppWidget Service
- Recognition Service
- DiskStats Service

ServiceManager



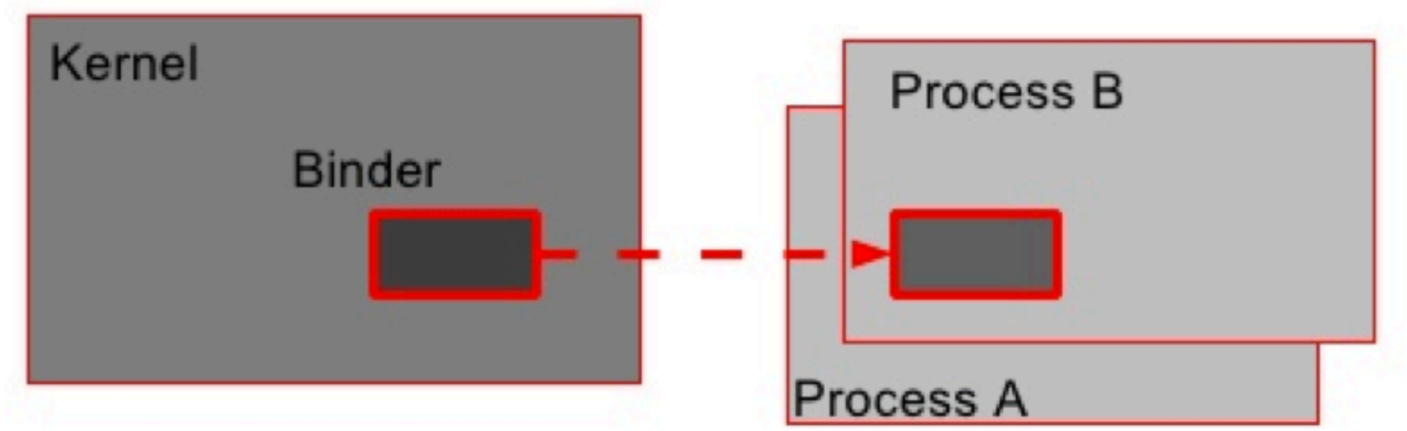

process A
service manager
Binder kernel
Service B
open binder driver
record Service B
open binder driver & record process A
add service
transfer data
& wakeup
suspend Service B
register service B
reply
wake up Service B
reply with status
get Service B
suspend process A
get service B
reply with result
wakeup process A
reply with Service B IBinder obj in kernel
transact
after connected with service B
reply
binder_loop
Thread pool
Client and Service connection flow

Binder Transactions

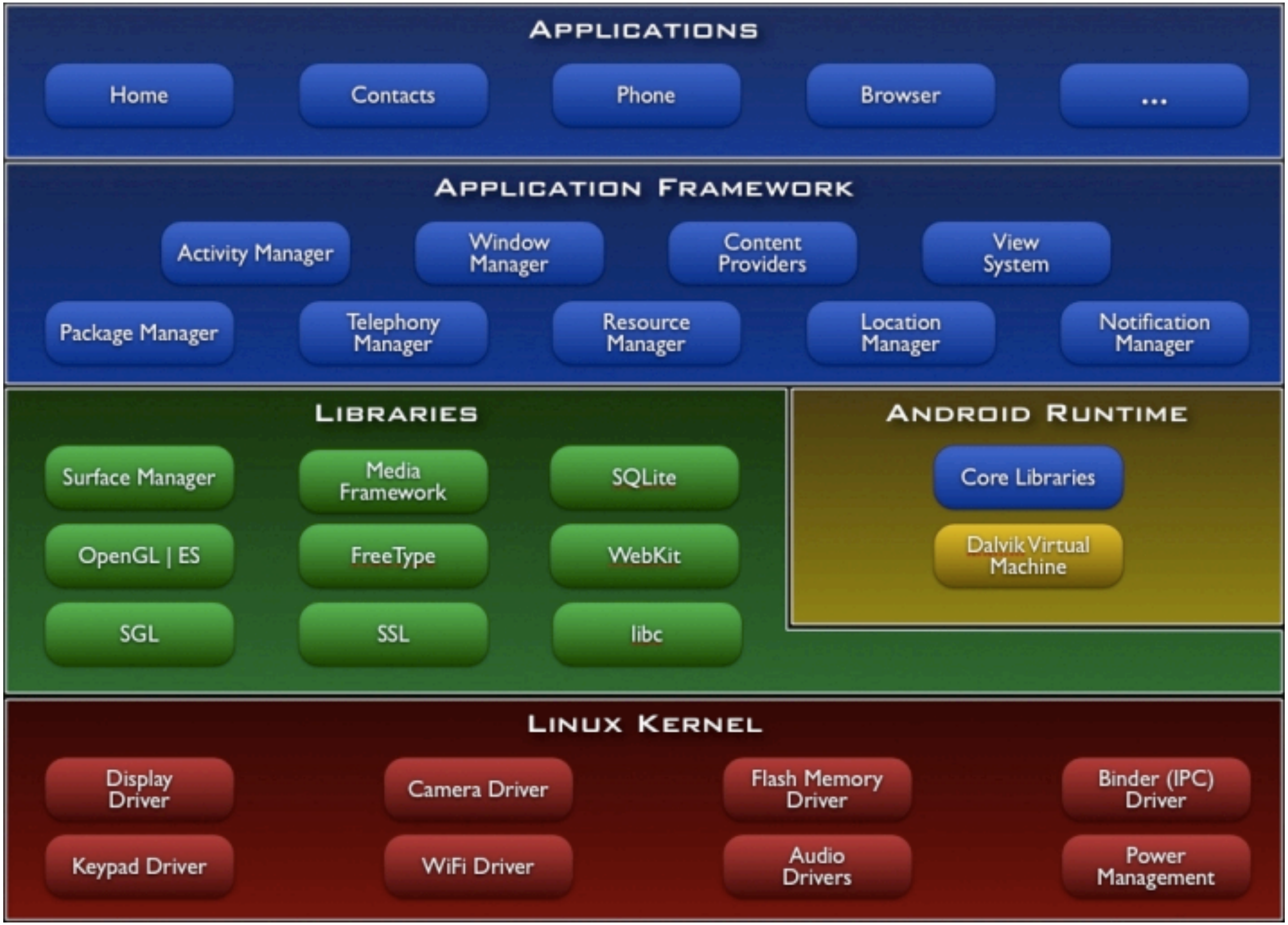


Copy memory by **copy_from _user**

Then, wake up process B



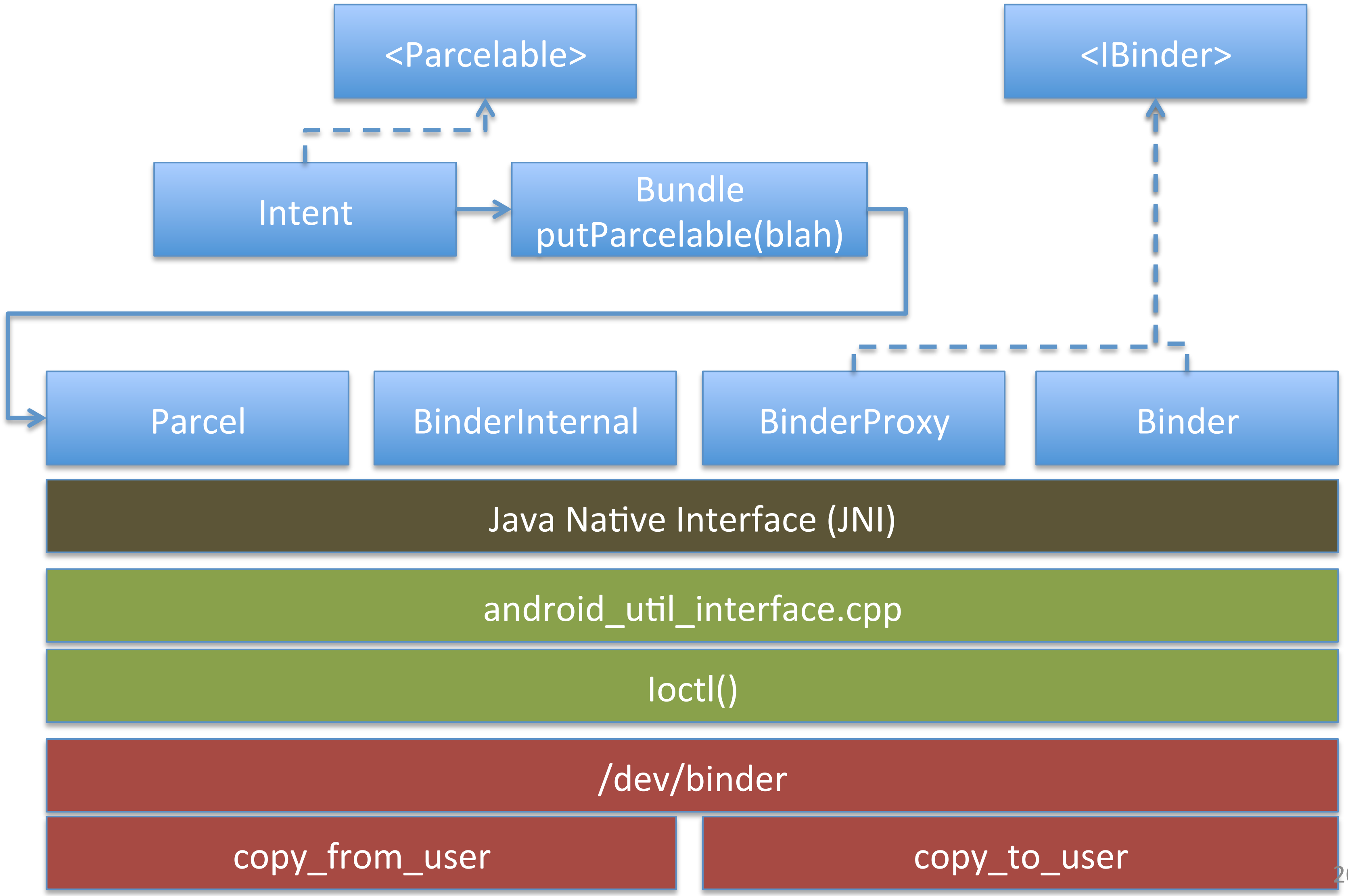
Copy memory by **copy_to_user**


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

Binder Implementation

- API for apps
  - Written in Java
  - AIDL
  - Java API wrapper
    - Exposes the IBinder interface
    - Wraps the middleware layer
    - Parcelable object marshalling interface
- Middleware
  - Written in C++
  - Implements the user space (i.e. within a process) facilities of the Binder framework
  - Marshalling and unmarshalling of specific data to primitives
  - Provides interaction with the Binder kernel driver
- Kernel drivers
  - Written in C
  - Supports ioctl system calls from the middleware
  - Supports cross-process file operations, memory mapping
  - Thread pool for each service application for IPC
  - Mapping of objects between processes via copy_from_user, copy_to_user

Binder Implementation



Binder Performance / Limitations

- Binders communicate over process boundaries
  - Processes do not share a common virtual machine context
    - No direct access to objects
  - Not ideal of large data-streams
    - i.e. audio/video
    - Parcelable overhead
  - Good enough for window / activity / surface management
- Advantages
  - Native binary marshalling
    - Not java serialisation
  - Support of ashmem shared memory
- Disadvantages
  - Overhead of Dalvik Parcel marshalling
  - Ioctl() not optimal
  - Passes file descriptors for faster binary data transfer

Binder Security

- Binder Security Features
  - Client identity managed by the kernel
    - Binder.getCallingUid(), Binder.getCallingPid()
  - Interface reference security
    - Client cannot guess “address” of a service without going via the Service Manager
- Service Manager
  - A directory service for system services
    - Mediate access
  - Revoke access based on token
- Service could check client permissions at run-time
  - Context.checkPermission(permission, pid, uid)

Services recap

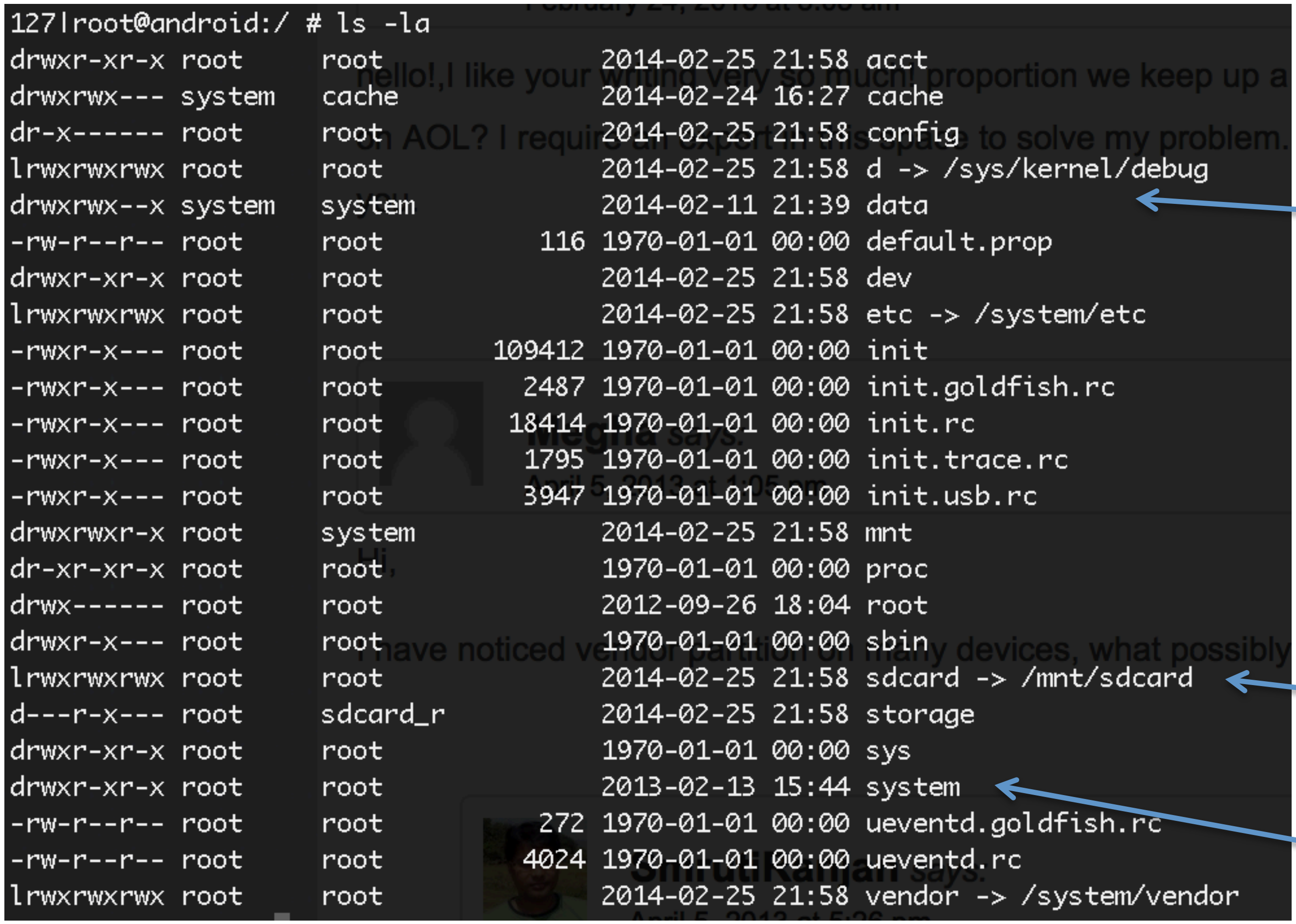
- A second kind of Android component
  - An abstraction of Binder / IPC
    - Used throughout the Android OS
- Tightly or loosely coupled to Activities
  - Start / destroy
    - Either by the Application
      - If we start it, it will run until we stop it
    - Or by the OS
      - If the OS starts it because it was bound, the OS destroys it when it is unbound
  - Communicate tightly via a Binder instance
    - Locally or remotely across processes
  - Communicate loosely via Notifications / Intents / Messages

References

- http://developer.android.com/guide/components/processes-and-threads.html
- http://developer.android.com/guide/components/services.html
- http://elinux.org/Android_Binder

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

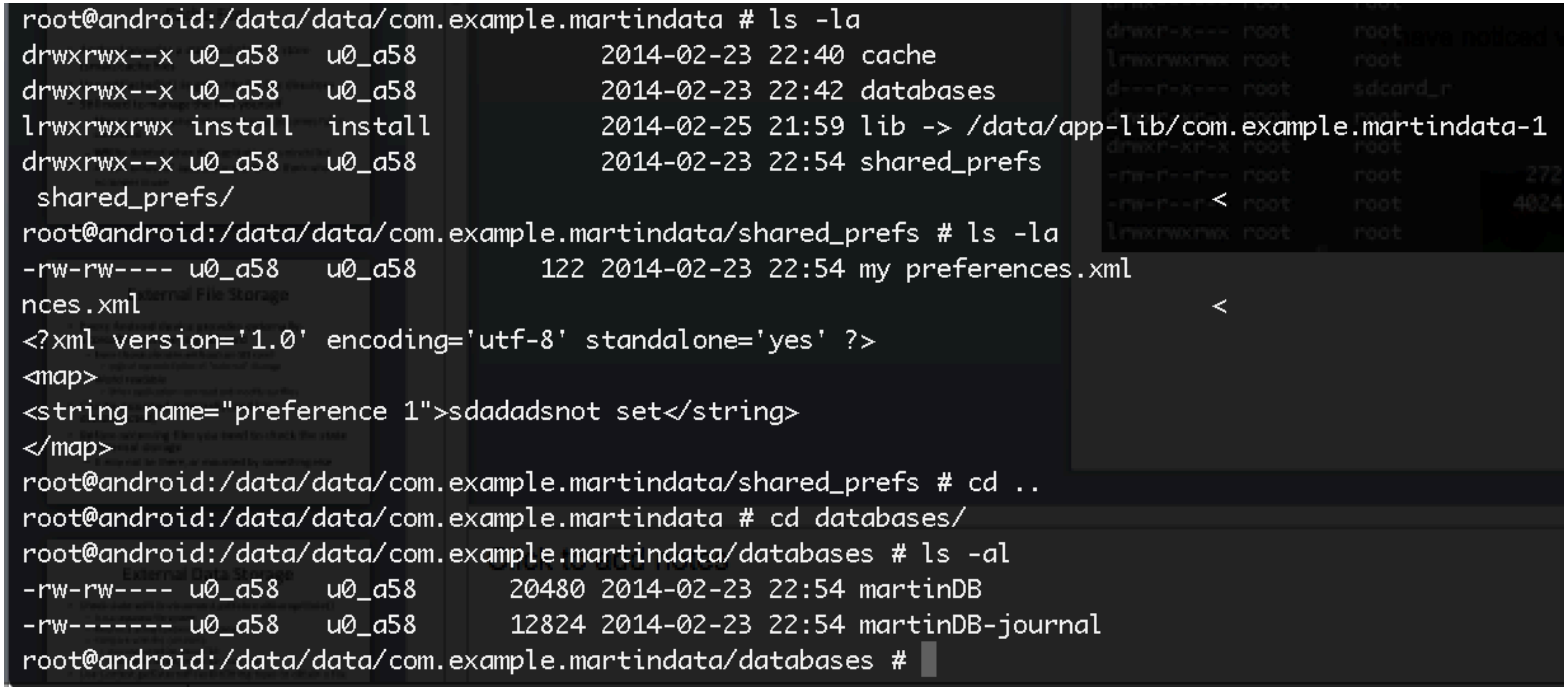
Android OS / libraries

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



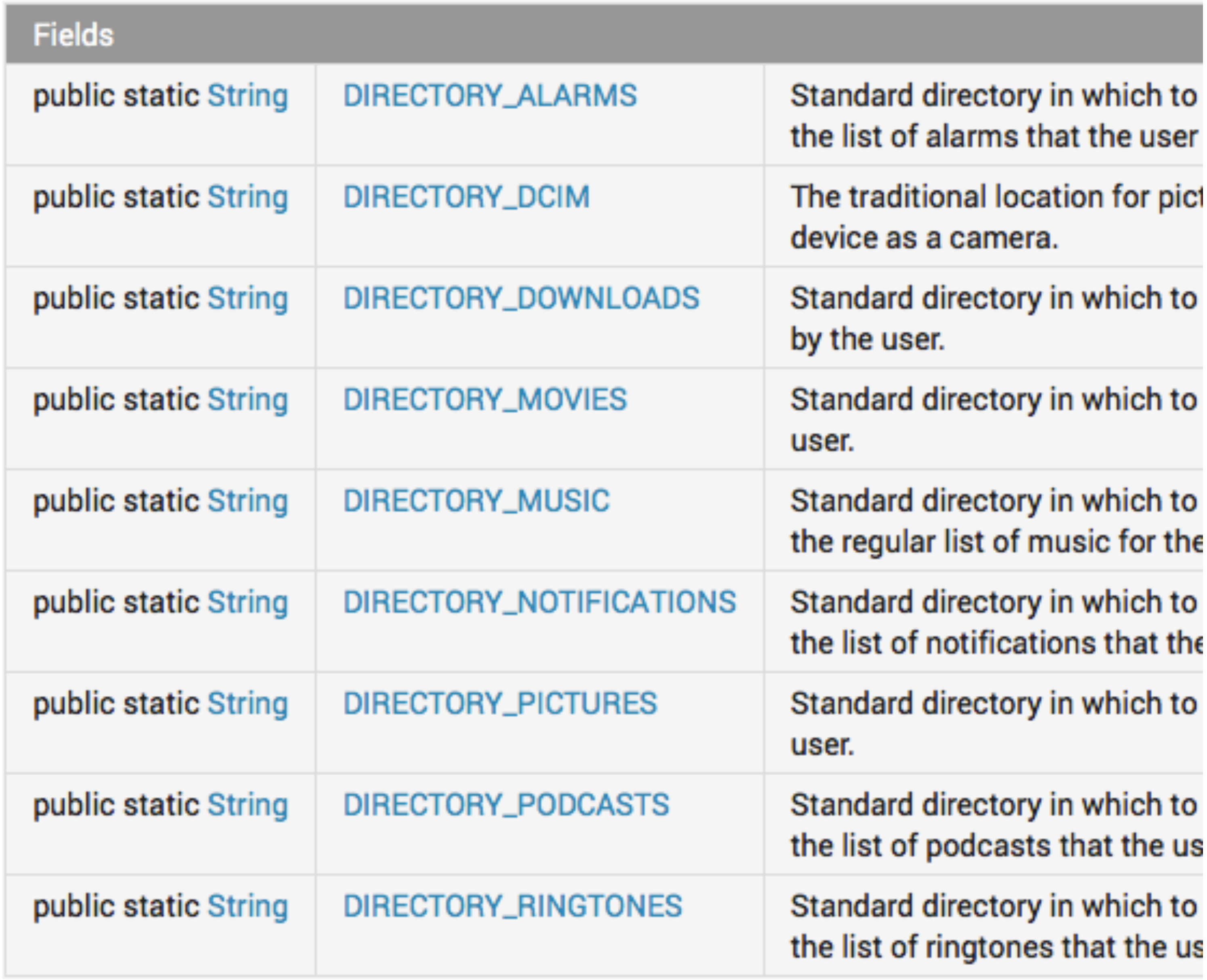
```
root@android:/data/data/com.example.martindata # ls -la
drwxrwx--x u0_a58   u0_a58            2014-02-23 22:40 cache
drwxrwx--x u0_a58   u0_a58            2014-02-23 22:42 databases
lrwxrwxrwx install  install           2014-02-25 21:59 lib -> /data/app-lib/com.example.martindata-1
drwxrwx--x u0_a58   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
  - World readable
    - Other applications can read and modify our 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 Context.getExternalFilesDir(String type) to obtain a File for the directory
  - 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()



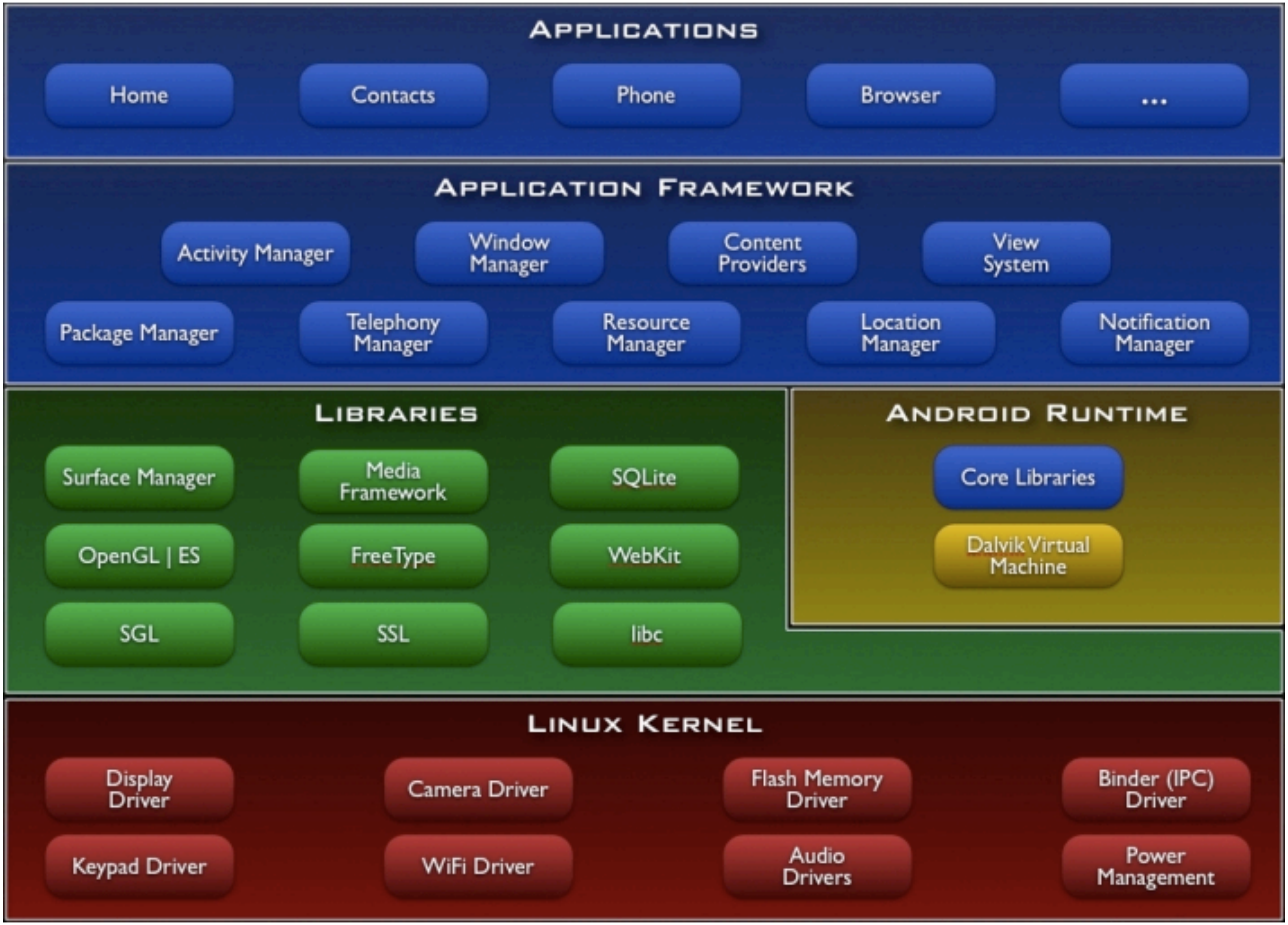
Fields		
public static String	DIRECTORY_ALARMS	Standard directory in which to the list of alarms that the user
public static String	DIRECTORY_DCIM	The traditional location for pict device as a camera.
public static String	DIRECTORY_DOWNLOADS	Standard directory in which to by the user.
public static String	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

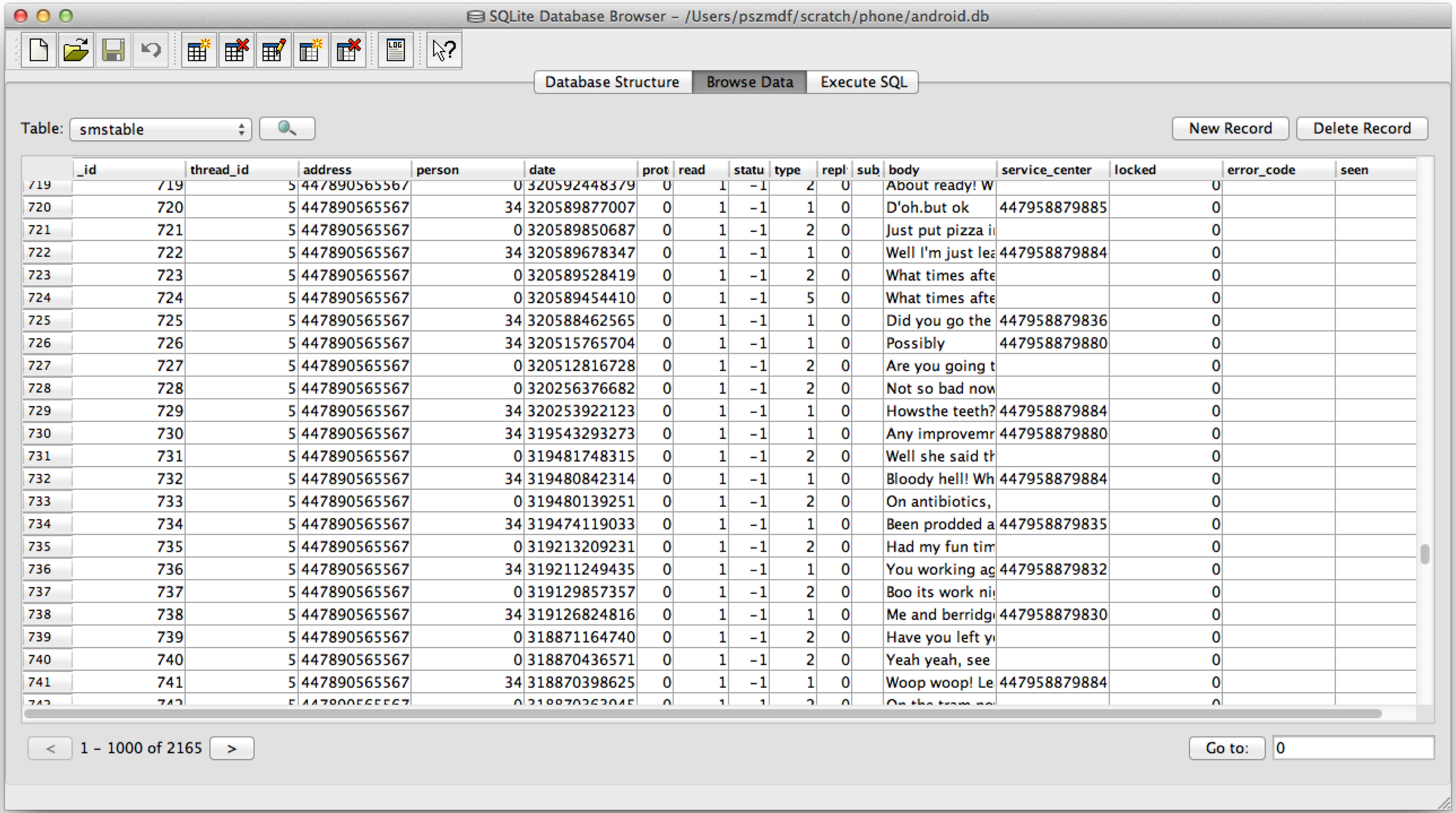
Structured Data

- 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



	_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 improvemr	447958879880	0		
731	731	5	447890565567	0	319481748315	0	1	-1	2	0		Well she said tl		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	[illegible]	0	1	-1	2	0		On the tram no		0		

1 – 1000 of 2165

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

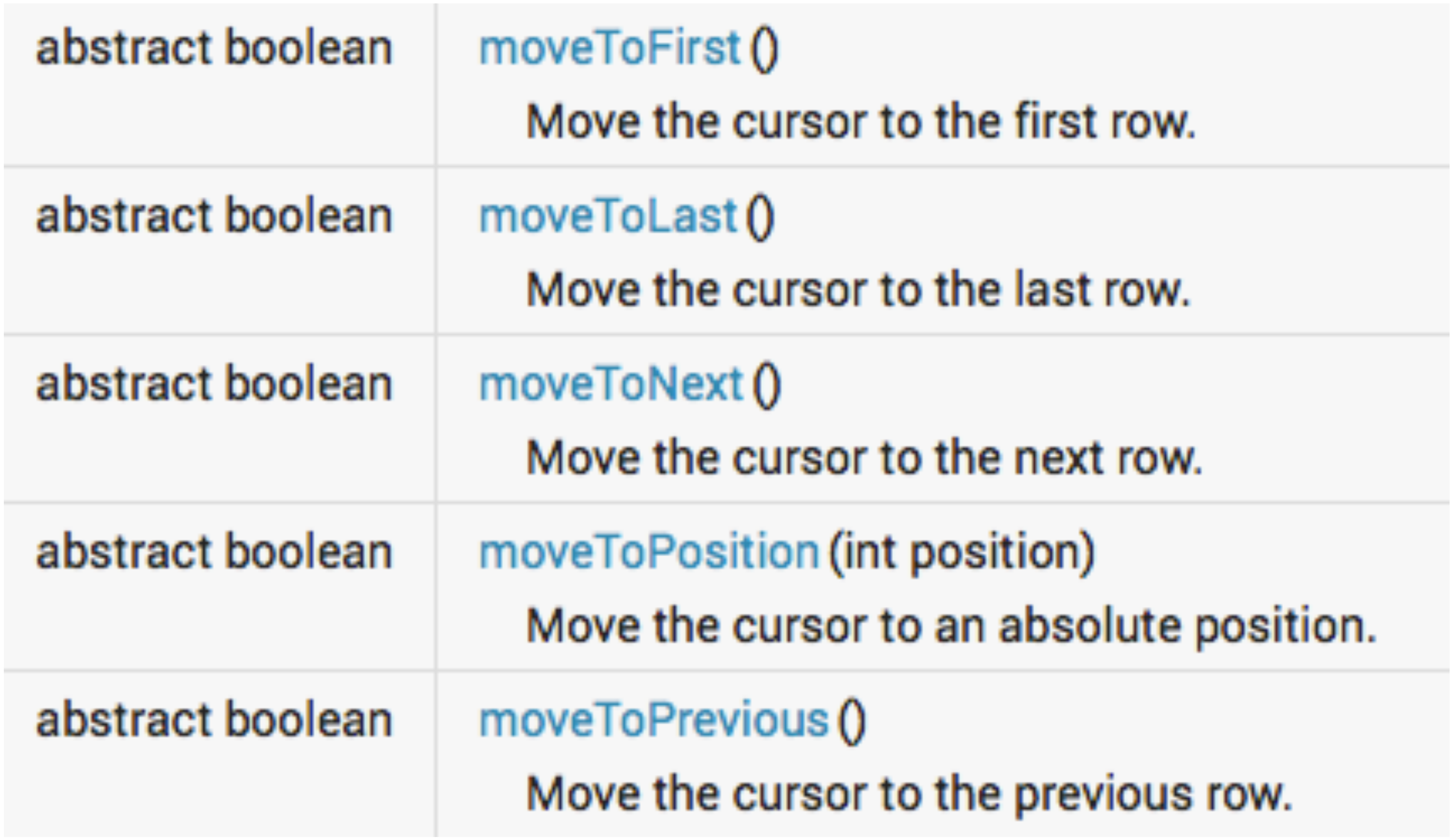
- SQLiteDatabase has many methods
- void execSQL()
  - used to run SQL queries that don’t return anything
- More useful are query() and rawQuery()
  - These return a Cursor object that can be used to access the data
  - “Move” the Cursor around the results
  - Provides random access to the results

Querying a Database

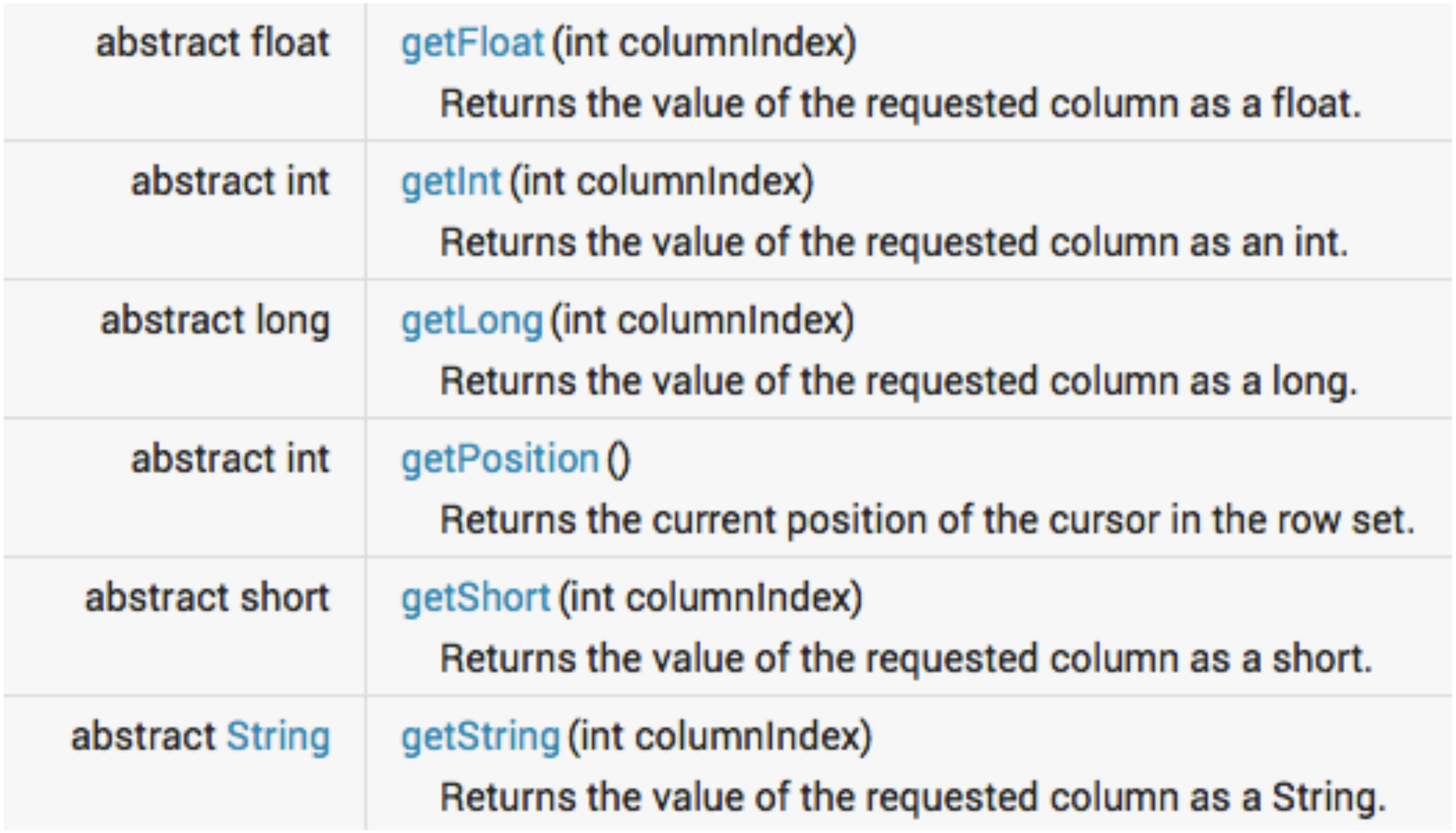
- `Cursor rawQuery(String sql, String[] selectionArgs)`
  - processes a raw SQL query
  - `rawQuery("SELECT id, name FROM people WHERE name = ? AND id = ?", new String[] {”Martin", ”78"});`
- SQL has to be parsed so there is also query() where the SQL is exploded into separate strings
  - Simpler to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



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.

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