G54MDP
Mobile Device Programming

Lecture 8 – Services / IPC Internals

Coursework

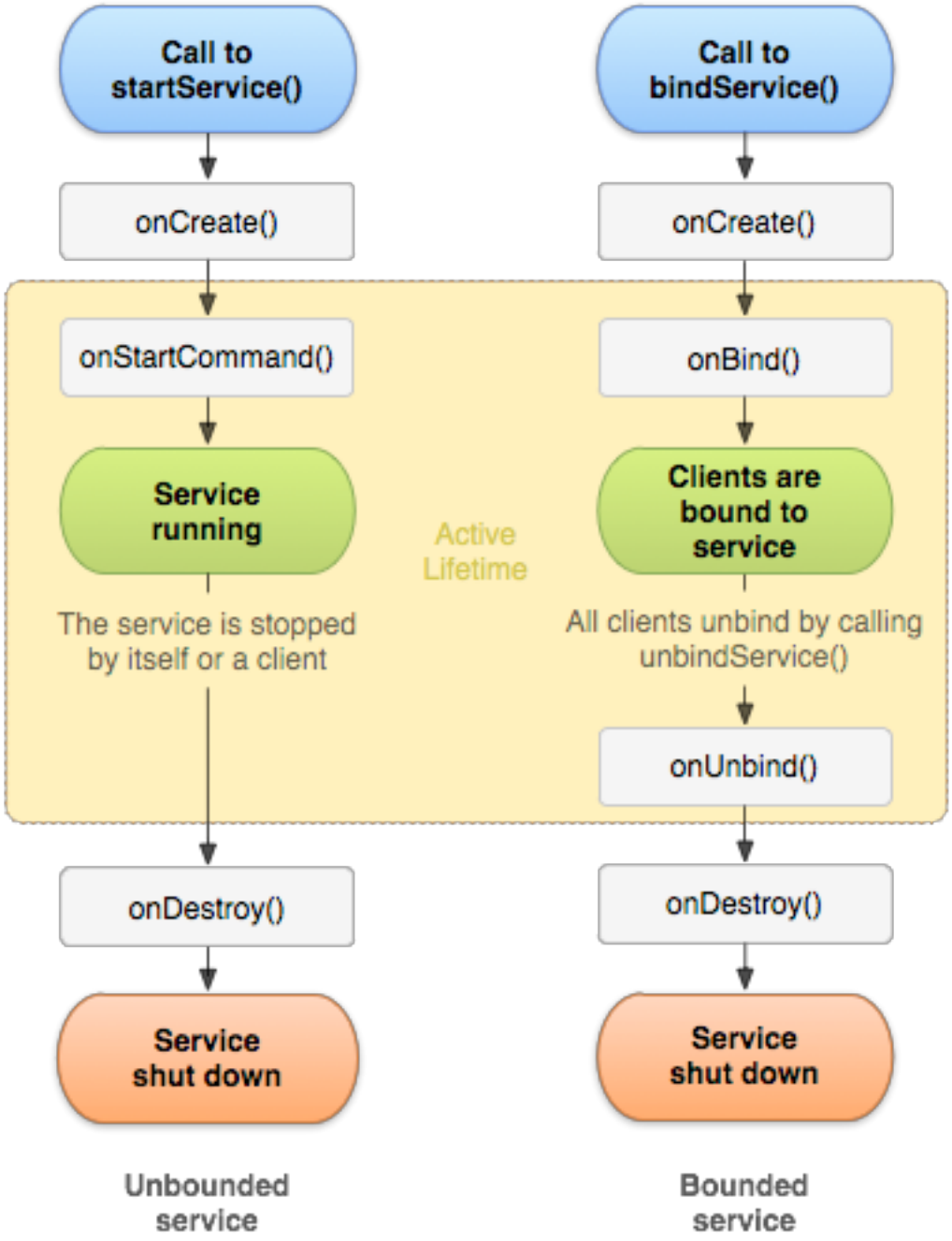
- Kitchen Egg Timer
- Due 4pm Wednesday 12th March
- Write code
  - No written component
- Activities and Services
  - Lifecycles
  - Decompose the task
  - Use appropriate views
- Don’t cheat

Feedback

- Whoops.
- G54MDP is not scheduled for Early Module Feedback
- However…
- Things I know about:
  - All lab exercises in advance
  - Early coursework
- What else?
  - (If you don't say anything I can't do anything about it)

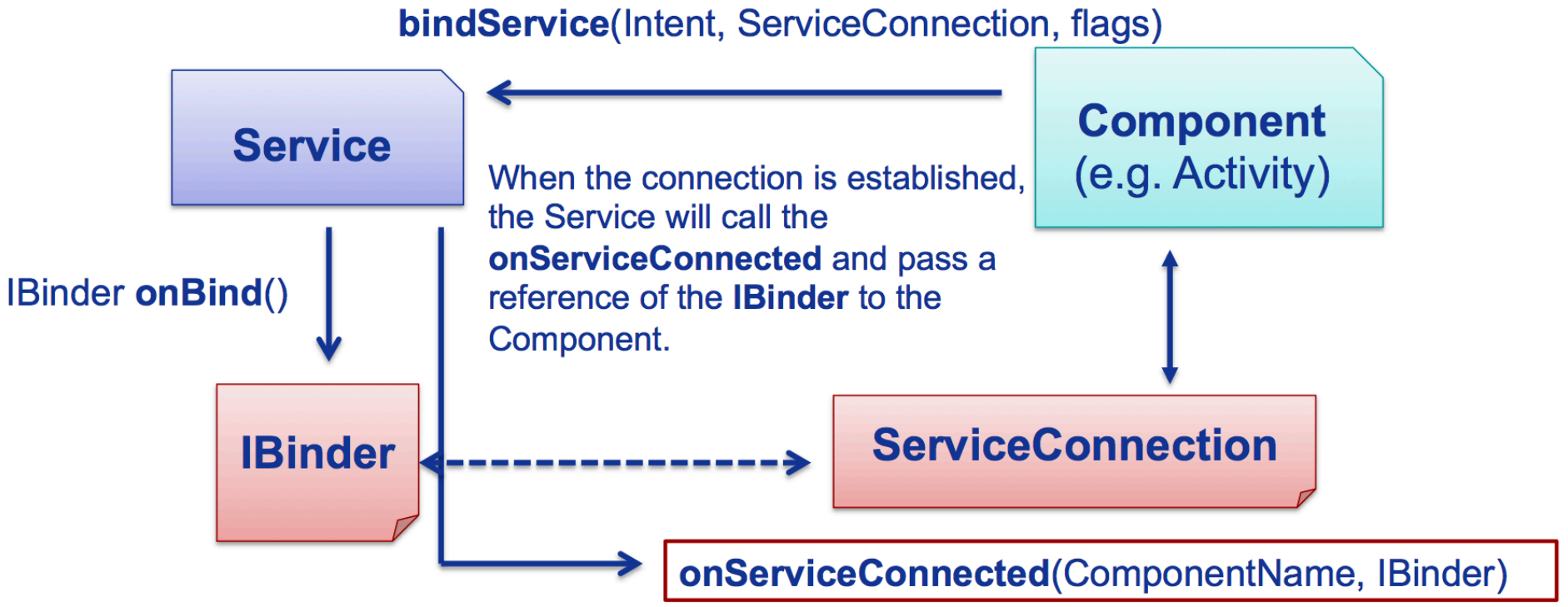
Service Lifecycle

- Two ways of spawning a service
  - Started
    - Send an Intent with startService()
    - Will run in the background indefinitely / kills itself
      - C.f email checking
      - Does not “return” results
    - Or can be explicitly stopped with stopService()
  - Bound
    - Bind to a service using bindService()
    - Will run while any Activities are bound to it
      - Actively using it
    - Provides an interface for Activities to communicate with the Service
- In both cases, if the service is not running it will be created


Call to startService()
onCreate()
onStartCommand()
Service running
The service is stopped by itself or a client
onDestroy()
Service shut down
Unbounded service
Call to bindService()
onCreate()
onBind()
Clients are bound to service
All clients unbind by calling unbindService()
onUnbind()
onDestroy()
Service shut down
Bounded service
Active Lifetime

Bound Services

- Provide an interface for clients (Activities) to interact with a Service
  - Provide a programmatic interface for clients
  - Fast *and* stable?
- **Extending** the Binder class
  - Return an interface via the onBind method
  - Only for a Service used by the same application
    - Local Services only
    - i.e. the same process
- **Using** the Android Interface Definition Language (AIDL)
  - Provide a standard interface to access the Service from different applications


bindService(Intent, ServiceConnection, flags)
Service
Component
(e.g. Activity)
When the connection is established, the Service will call the onServiceConnected and pass a reference of the IBinder to the Component.
IBinder onBind()
IBinder
ServiceConnection
onServiceConnected(ComponentName, IBinder)

Remote Bound Services

- For communicating across process boundaries
  - i.e. using a Service belonging to a different application / process
  - Likely to be used by multiple processes at once
- Using a Messenger
  - Simplest implementation
  - C.f. using a Handler to talk between Threads
    - Queues Messages into a single Thread
  - Messages must be Parcelable

Parcelable

- Locally (same process) bound Services share the same process memory space
  - Easy to call methods, transfer objects between classes
- How should different processes talk to each other?
  - java.io.Serializable
    - Short-term persistence
    - Write object ID, field via reflection
    - Change the class / variable name, what happens?
    - Slow
  - Parcelable
    - Define a simple wire-protocol for writing primitives
    - Immune to minor changes to class definitions
      - Same interface, different class
    - Supported by Android kernel driver
    - Fast!

Remotely Bound Services

- Define remote interface in the Android Interface Definition Language (AIDL)
  - Providing OS wide services for all applications
    - i.e download management
  - Multithreading with complex client / server bi-directional communication
- Implement remote interface
  - Stub and application specific methods
- Implement Service methods
- Implement Client methods

AIDL

- Similar to Java interface definition syntax
  - Can declare methods
  - Cannot declare static fields
- Label method parameters
  - in: transferred to the remote method
  - out: returned to the caller
  - inout: both in and out
- Types
  - Java **primitive** types
  - StringList
    - List elements must be valid AIDL data types
  - Map
    - Map elements must be valid AIDL data types
  - CharSequence
  - Other AIDL-generated interfaces
  - Classes implementing the Parcelable protocol

AIDL

- Generate a Java interface with same name as .aidl file Eclipse does this automatically
- Generated interface contains:
  - Abstract inner class called Stub
  - Interface & helper methods

Let’s have a look…

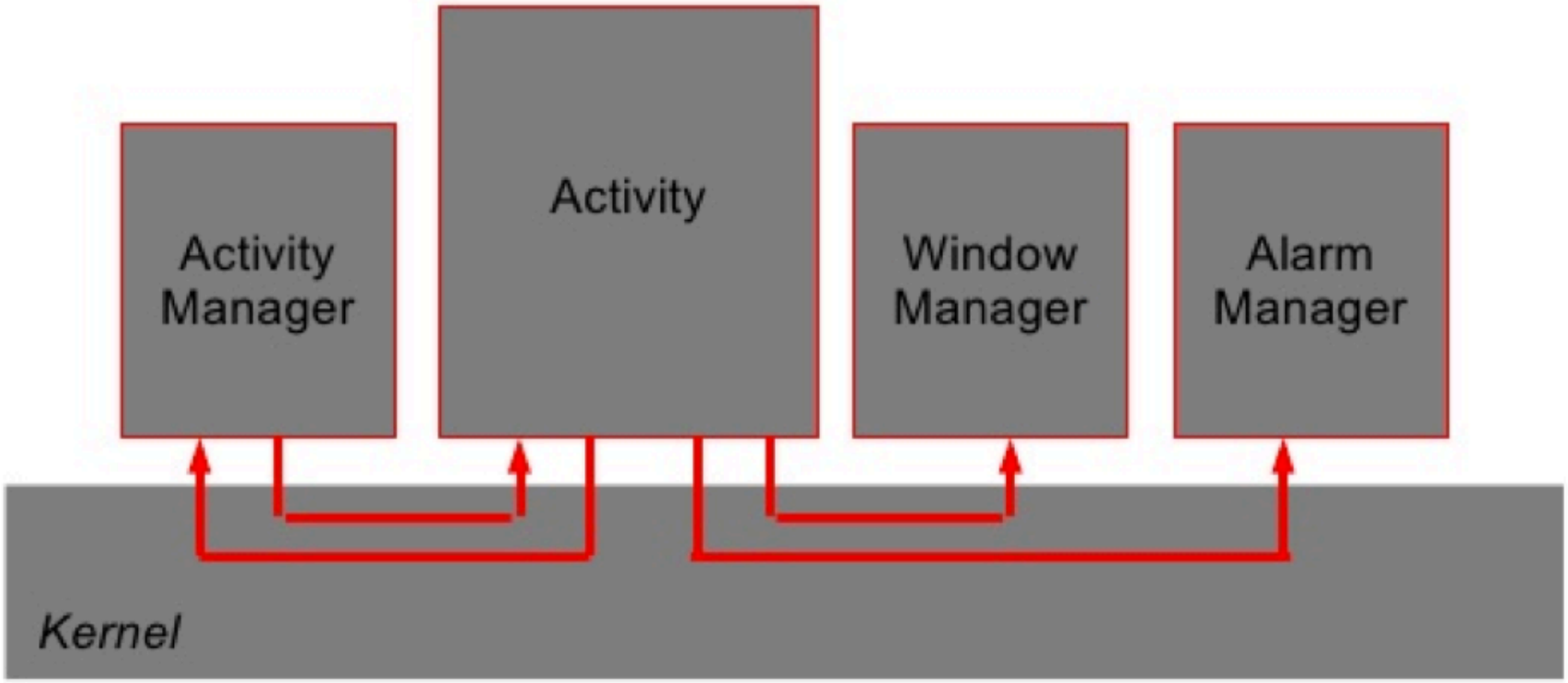
```
root@android:/ # service list
Found 68 services:
0       phone: [com.android.internal.telephony.ITelephony]
1       iphonesubinfo: [com.android.internal.telephony.IPhoneSubInfo]
2       simphonebook: [com.android.internal.telephony.IIccPhoneBook]
3       isms: [com.android.internal.telephony.ISms]
4       dreams: [android.service.dreams.IDreamManager]
5       commontime_management: []
6       samplingprofiler: []
7       diskstats: []
8       appwidget: [com.android.internal.appwidget.IAppWidgetService]
9       backup: [android.app.backup.IBackupManager]
10      uimode: [android.app.IUiModeManager]
11      serial: [android.hardware.ISerialManager]
12      usb: [android.hardware.usb.IUsbManager]
13      audio: [android.media.IAudioService]
14      wallpaper: [android.app.IWallpaperManager]
15      dropbox: [com.android.internal.os.IDropBoxManagerService]
16      search: [android.app.ISearchManager]
17      country_detector: [android.location.ICountryDetector]
```

```
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
- NetworkMnagement 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

IPC – Inter-Process Communication

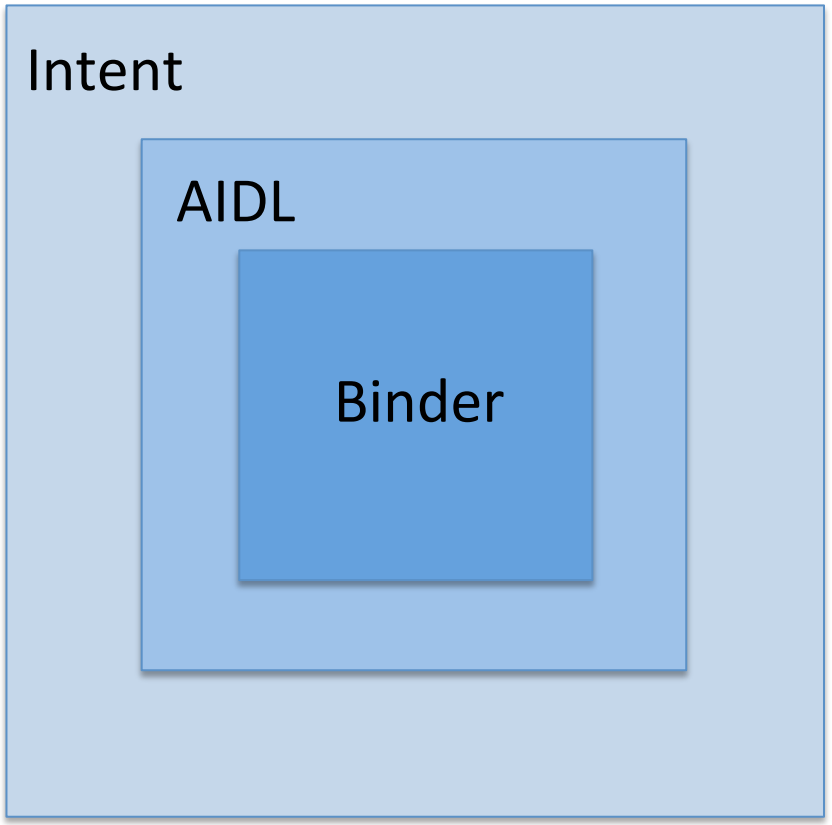


IPC

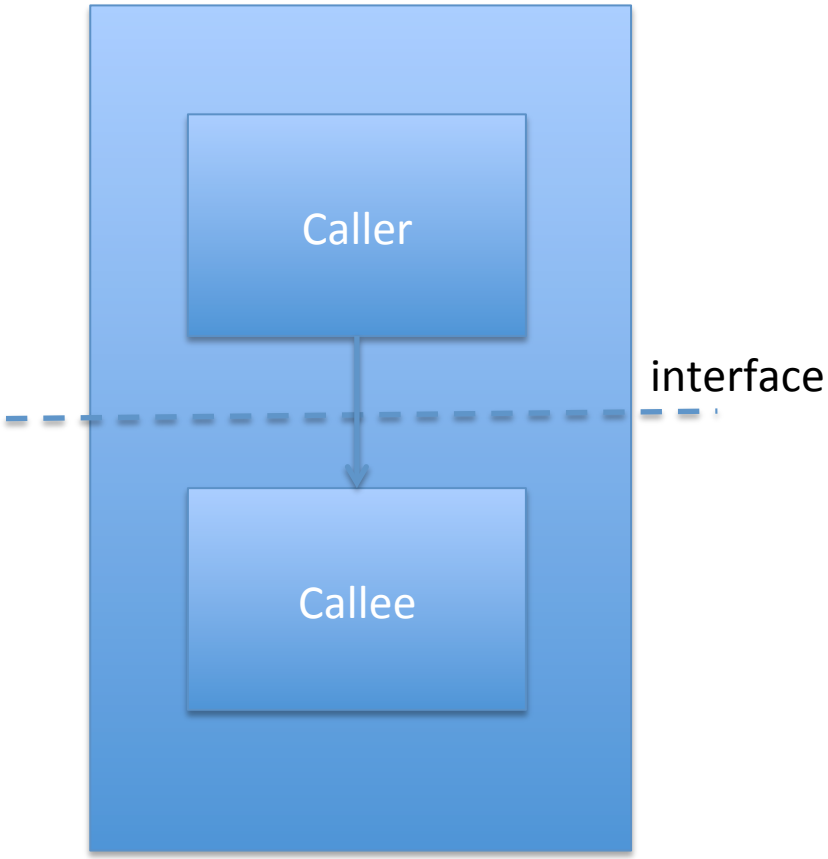
- Each process has its own address space
  - Provides data isolation
  - Prevents direct interaction between different processes
    - However, often required for modularisation
- What **actually** happens when we start a Service, or send an Intent?
- Binder
  - Underpins most Android communication
    - 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

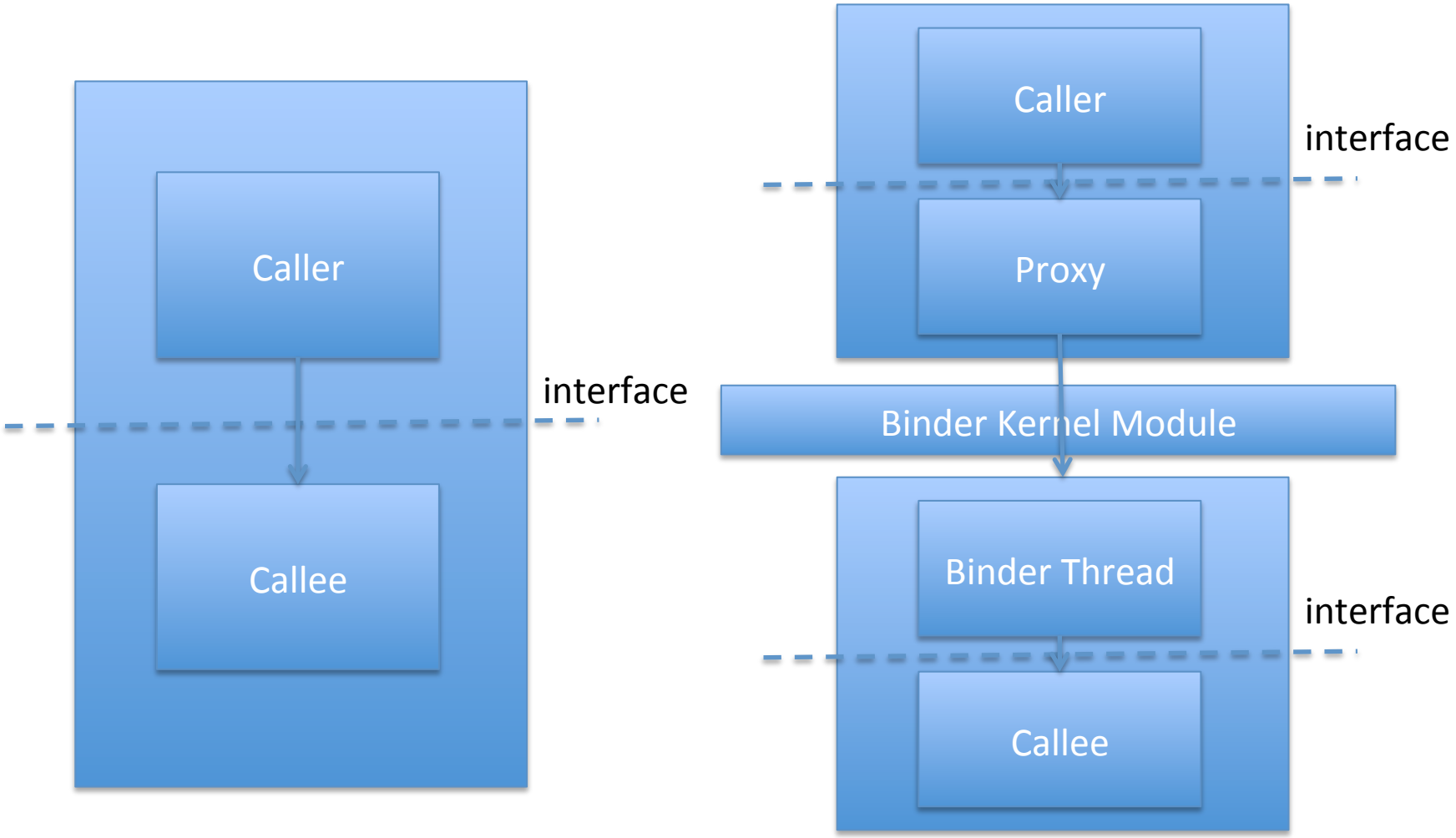
- Intent
  - Highest level abstraction
- Inter process method invocation
  - AIDL
- binder: kernel driver
- ashmem: shared memory



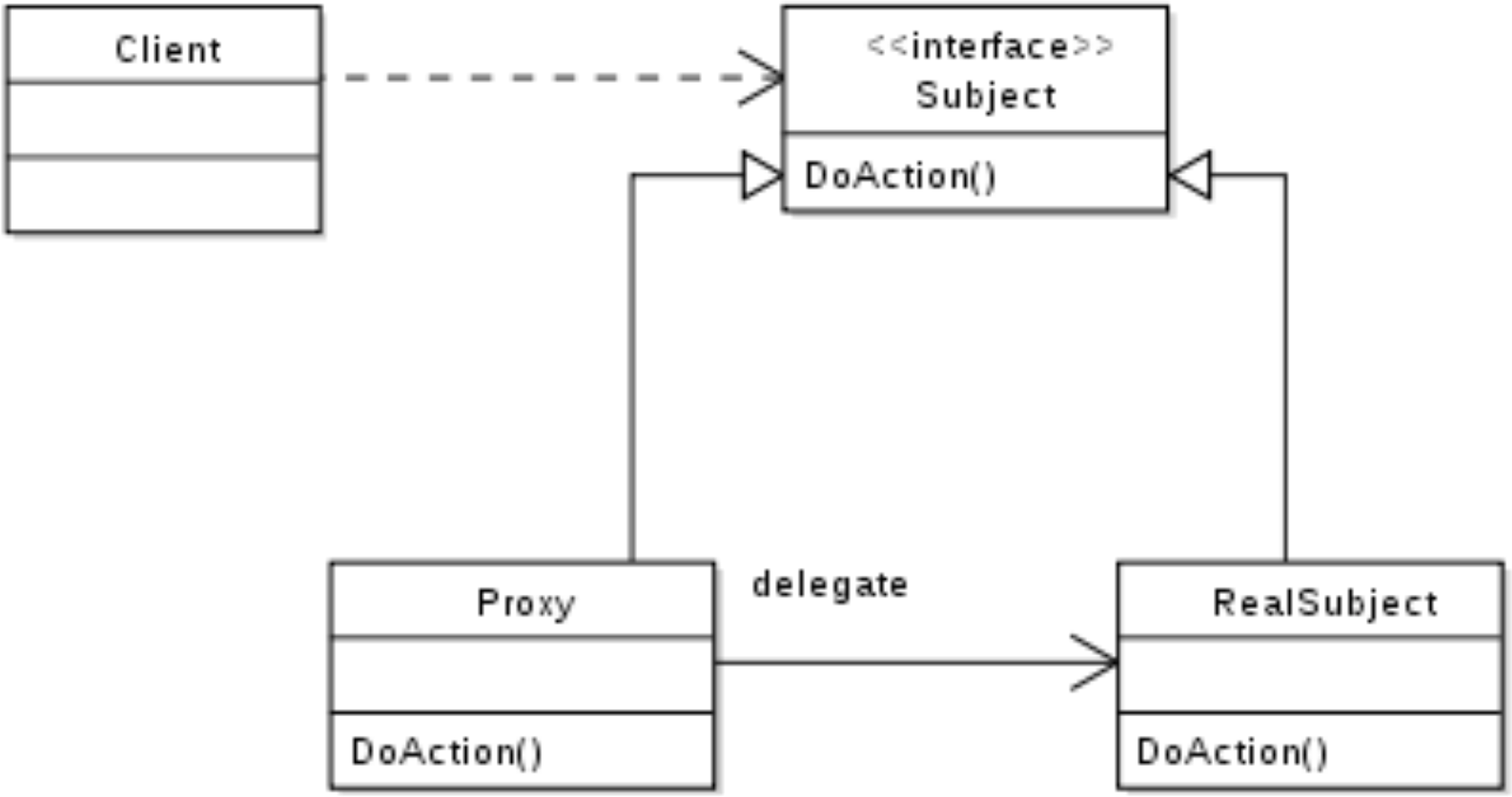
Inter-process method invocation



Inter-process method invocation

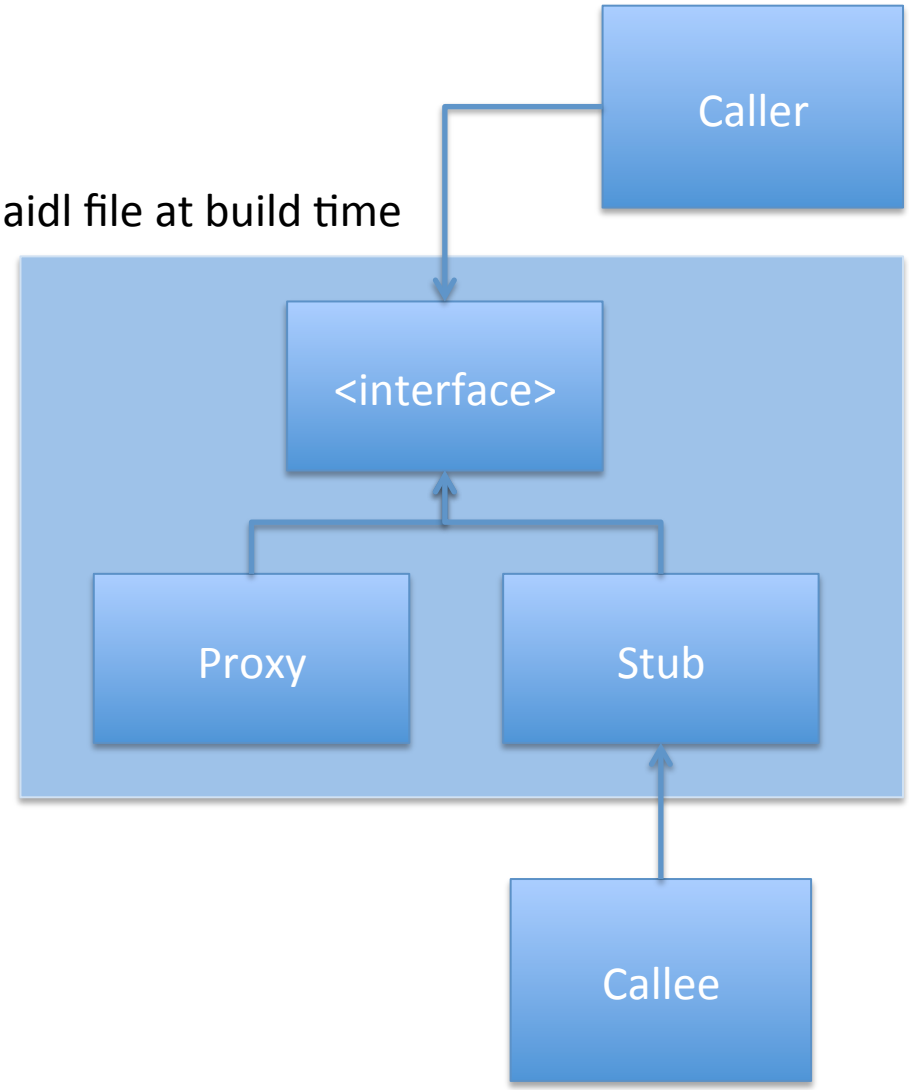


Proxy Design Pattern

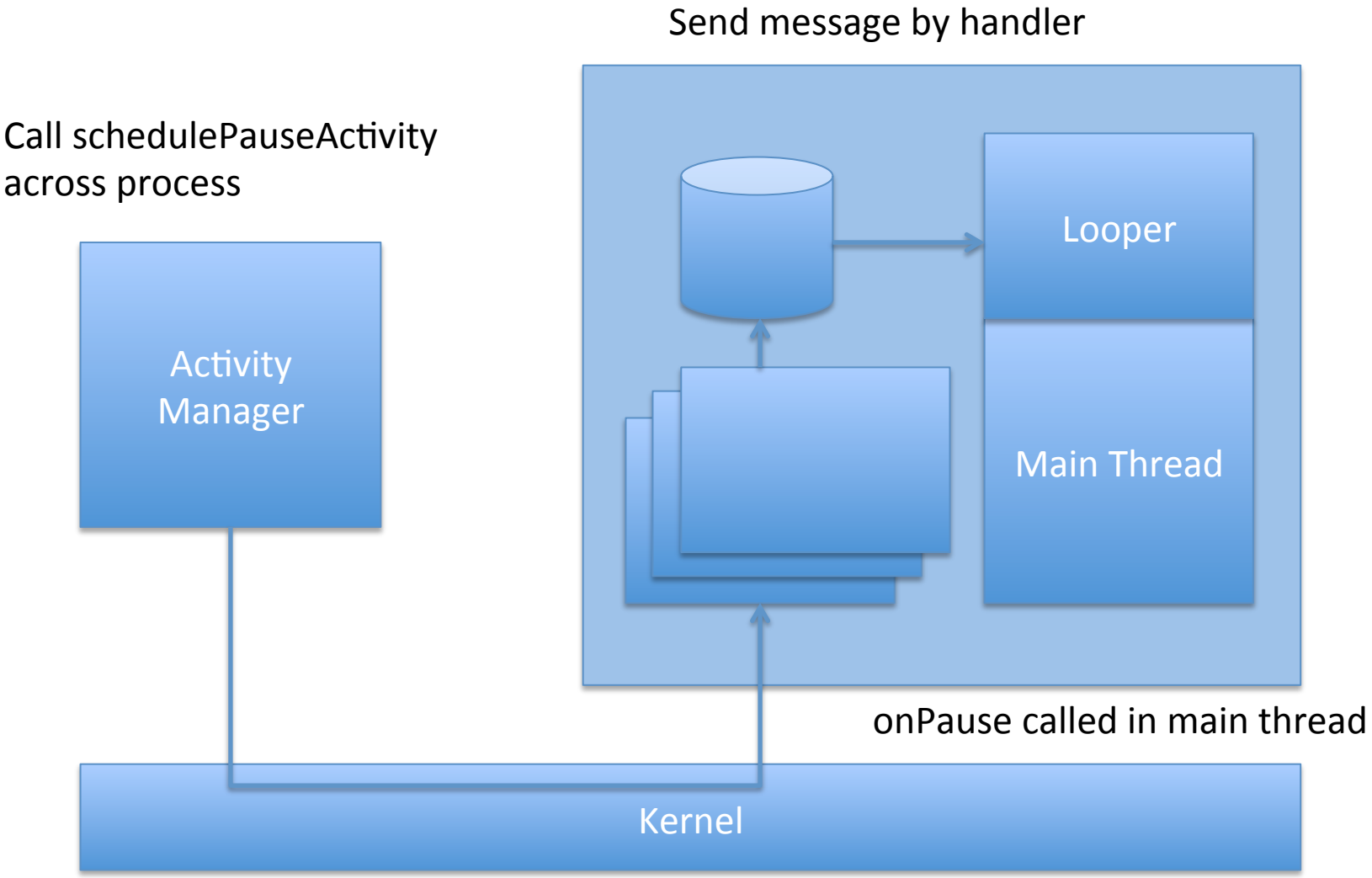


AIDL

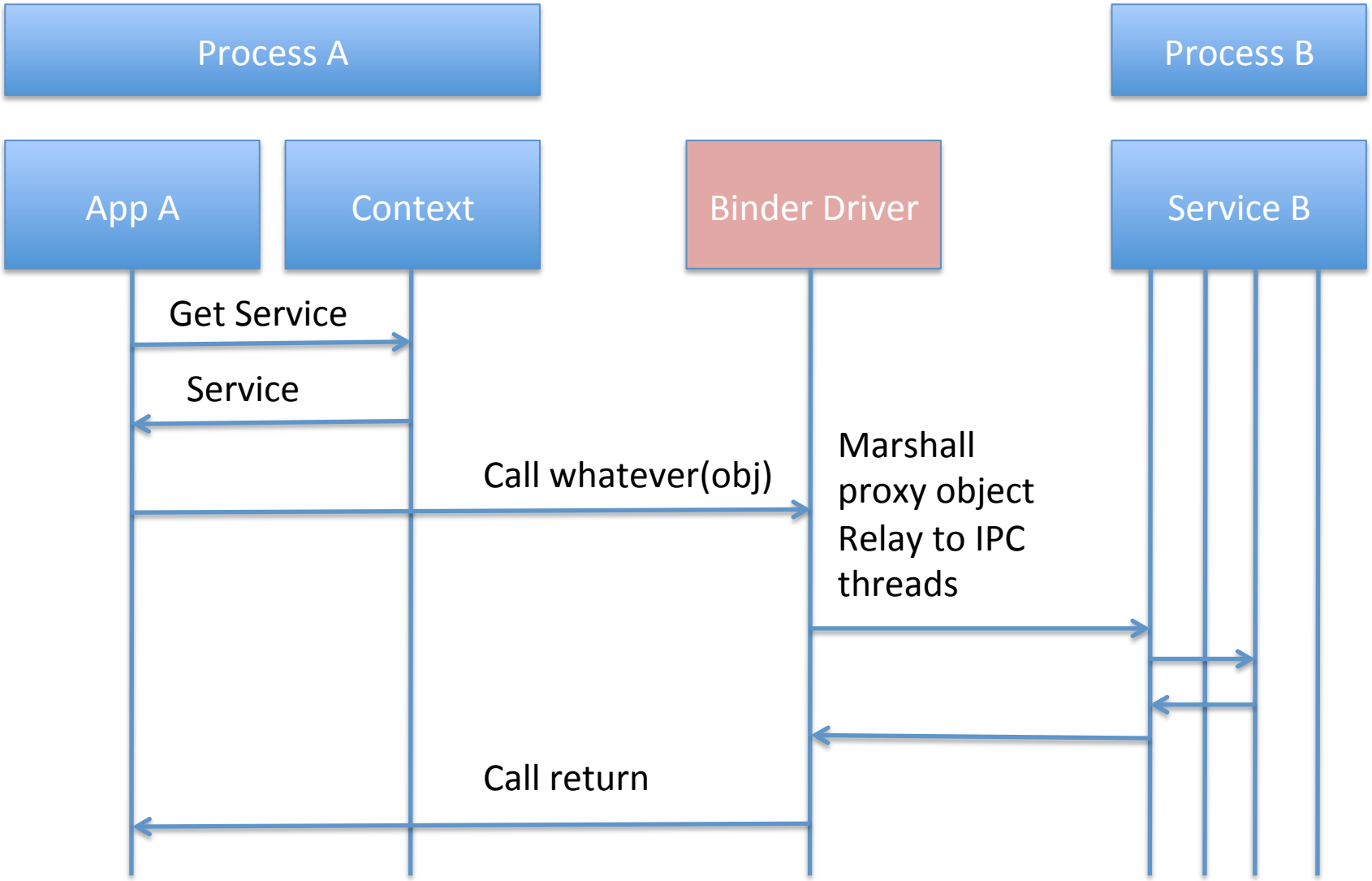
Auto generated from .aidl file at build time



onPause()


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

Binder in action



References

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