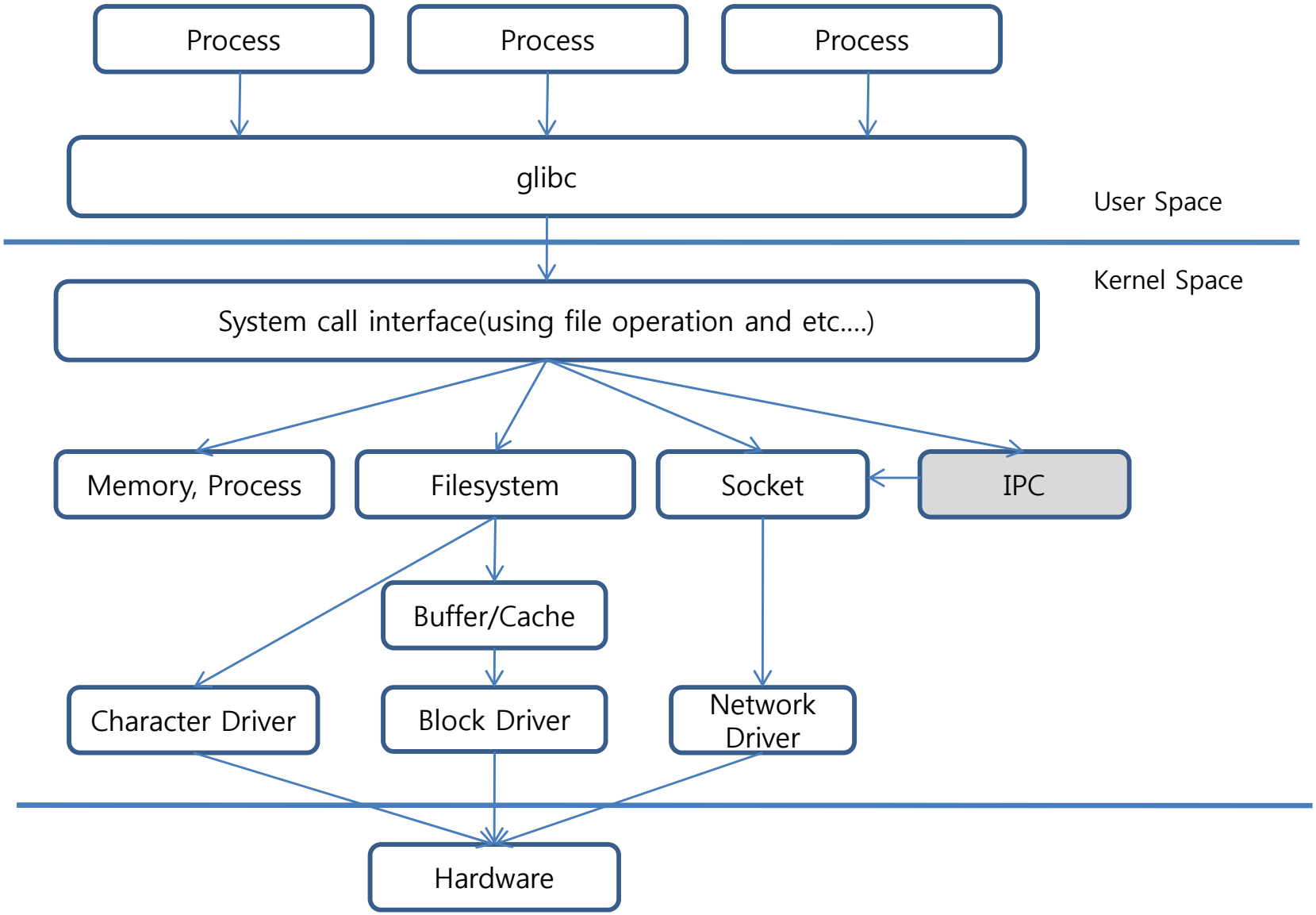
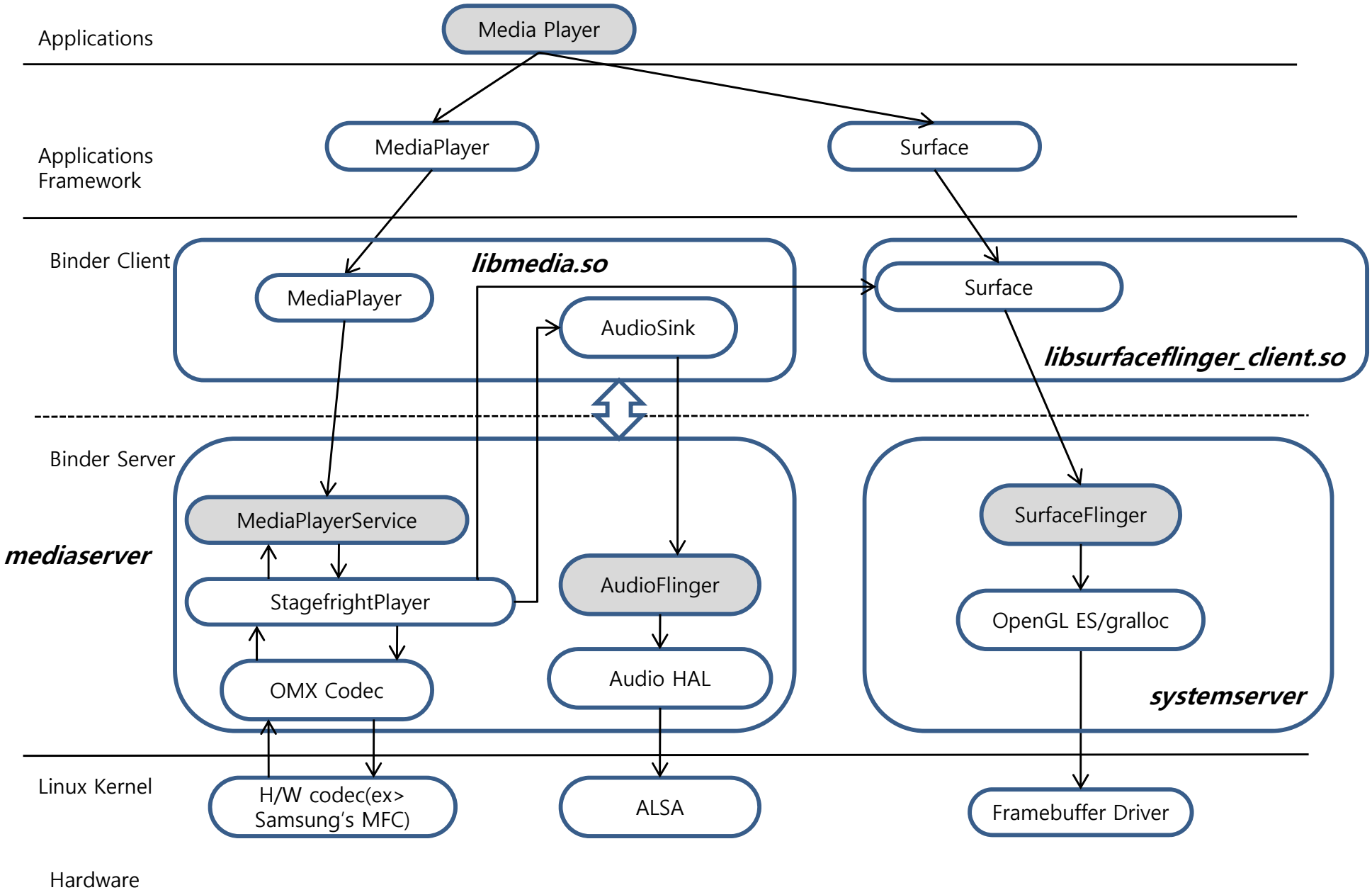
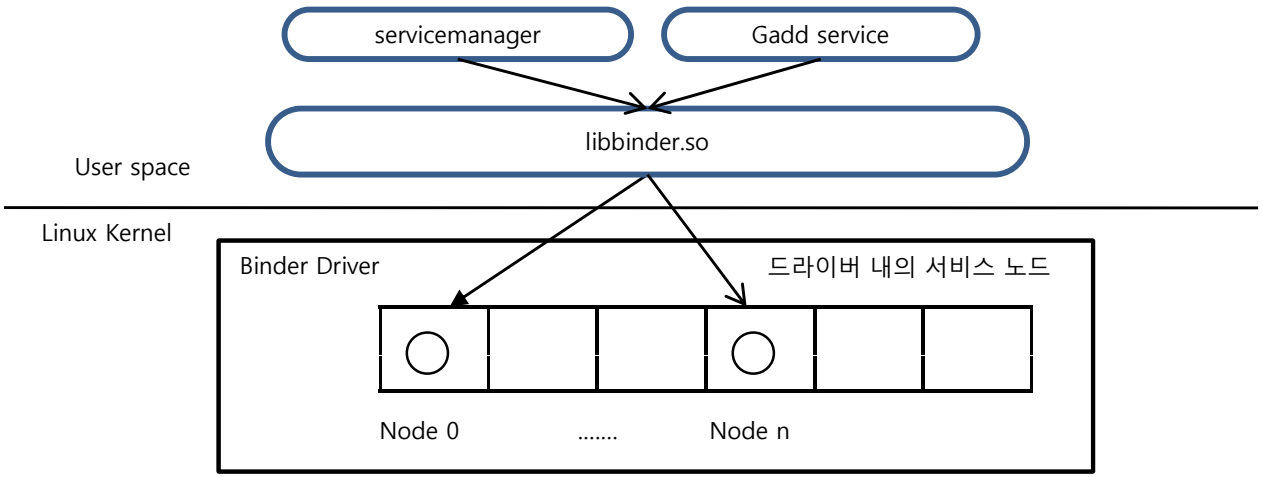
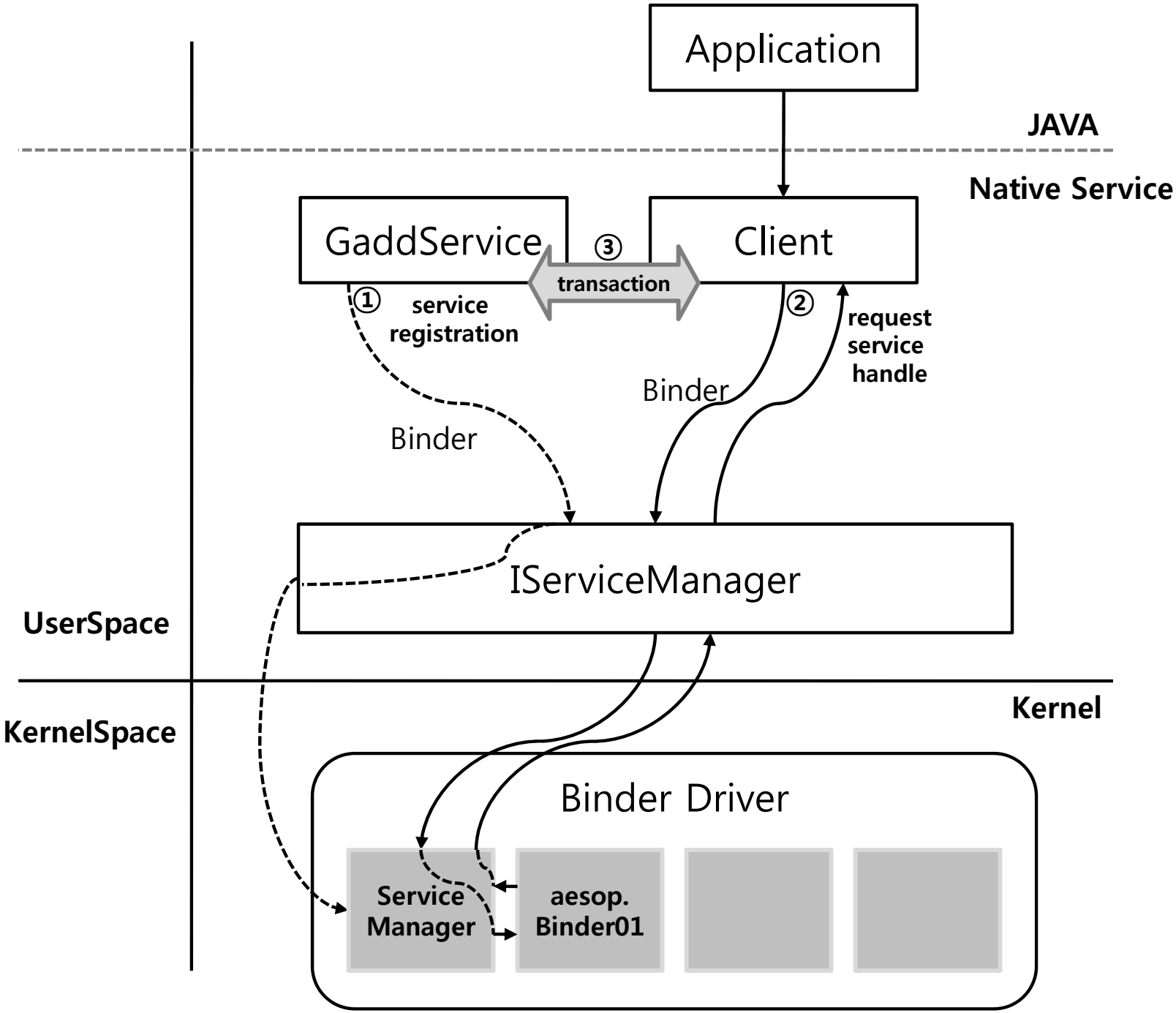
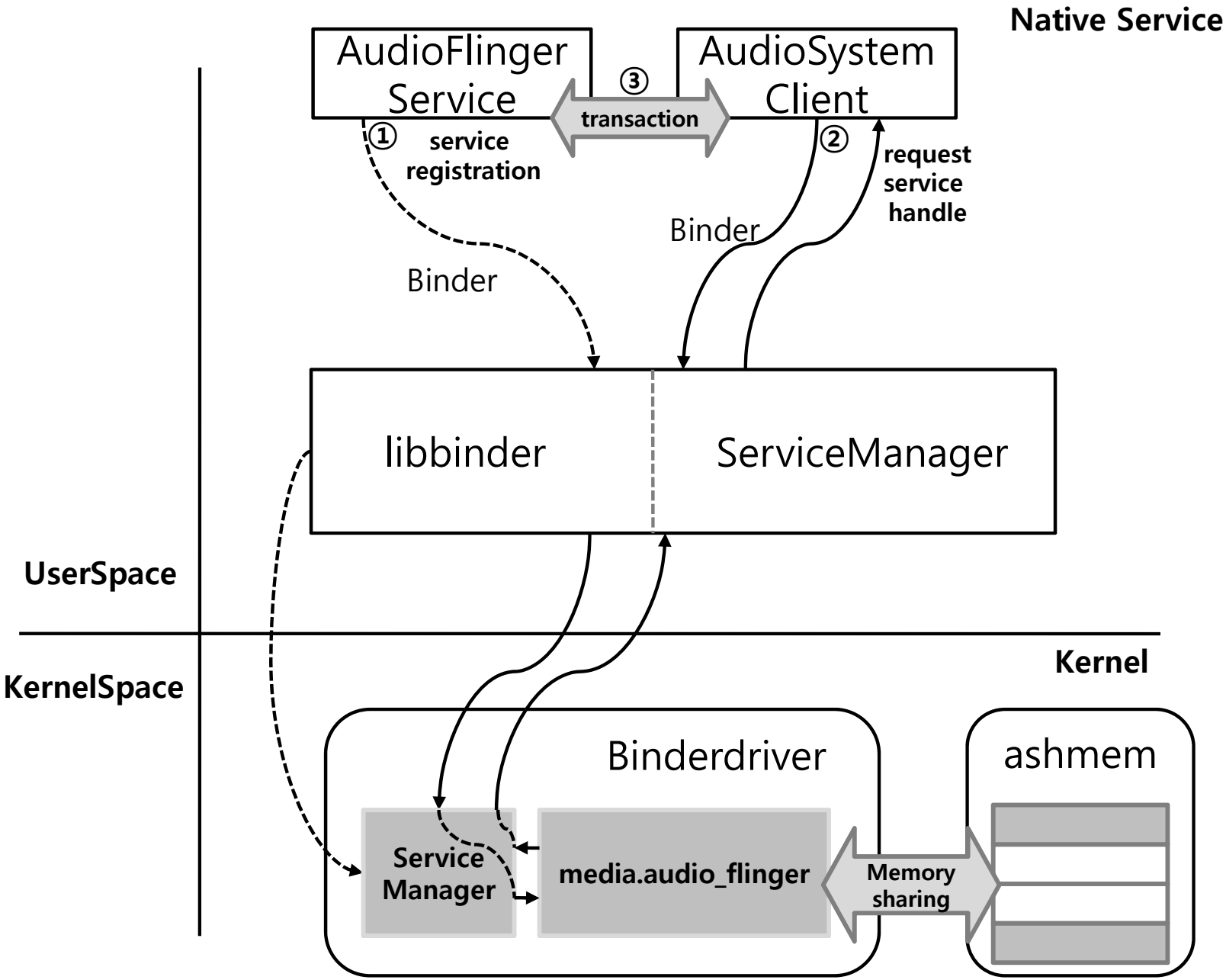

Process
Process
Process
glibc
User Space
Kernel Space
System call interface(using file operation and etc....)
Memory, Process
Filesystem
Socket
IPC
Buffer/Cache
Character Driver
Block Driver
Network Driver
Hardware

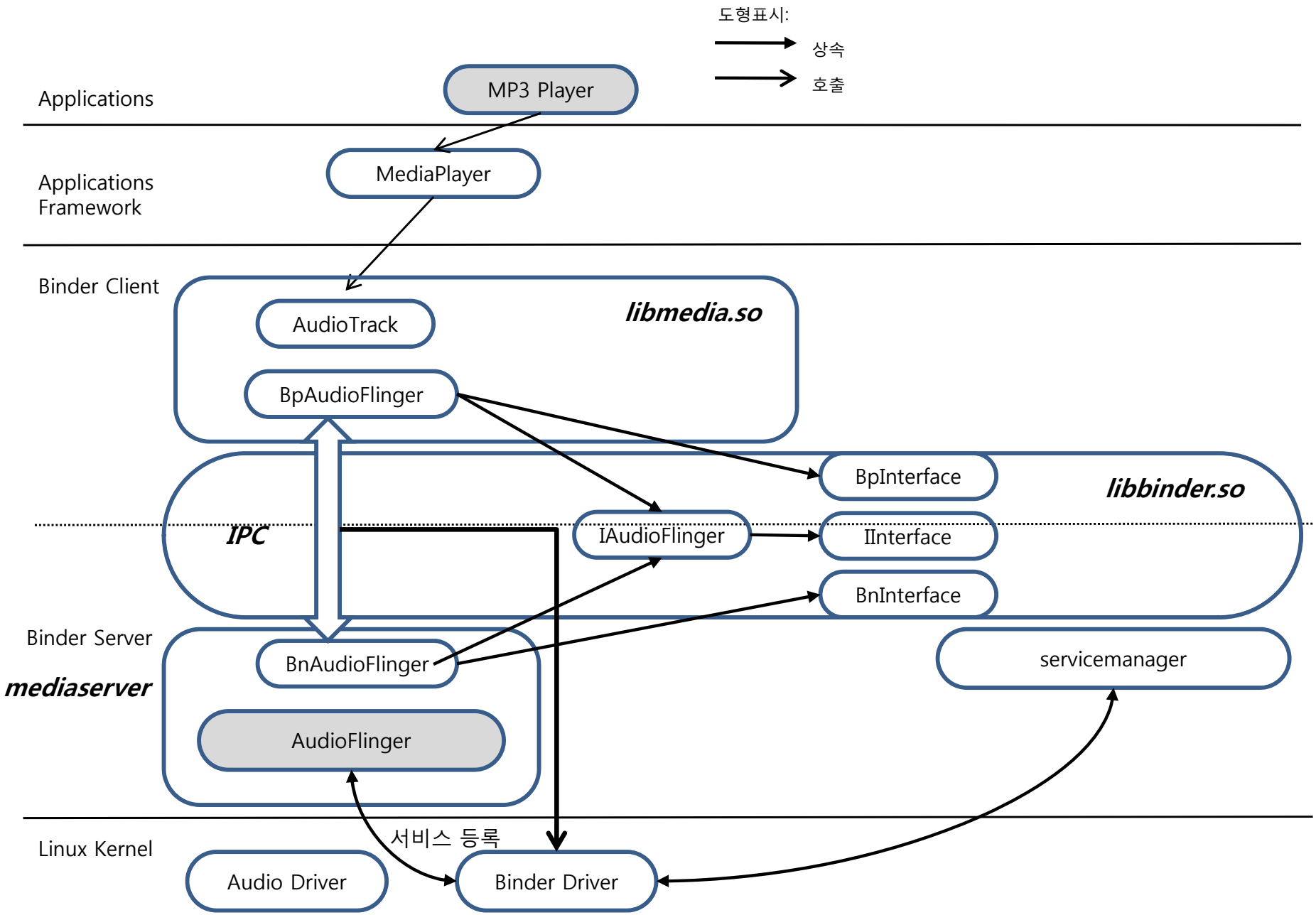

Applications
Media Player
Applications Framework
MediaPlayer
Surface
Binder Client
libmedia.so
MediaPlayer
AudioSink
Surface
libsurfaceflinger_client.so
Binder Server
mediaserver
MediaPlayerService
StagefrightPlayer
OMX Codec
AudioFlinger
Audio HAL
SurfaceFlinger
OpenGL ES/gralloc
systemserver
Linux Kernel
H/W codec(ex> Samsung's MFC)
ALSA
Framebuffer Driver
Hardware

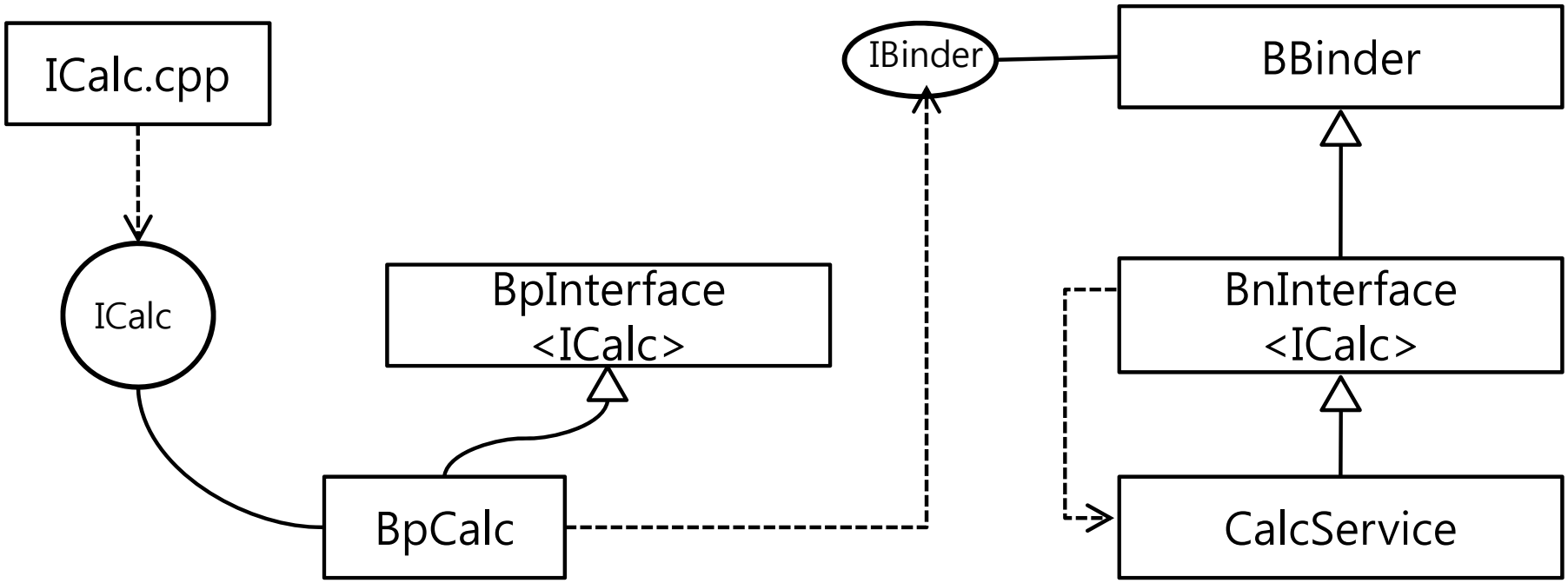
utils/RefBase.h **class RefBase**	바인더의 기본이 되는 class, 주로 참조 카운트 관련 클래스
binder/IBinder.h **class IBinder : public virtual RefBase**	바인더통신 관련 기본 클래스
binder/Binder.h **class BBinder : public IBinder**	IPC 용도로 사용할 경우 주로 사용되는 클래스
binder/IInterface.h **class IInterface : public virtual RefBase** **class BnInterface : public INTERFACE, public BBinder** **class BnInterface : public INTERFACE, public BBinder**	RPC 용도로 사용할 경우 주로 사용되는 클래스 BnInterface는 바인더 서버용 BpInterface는 바인더 클라이언트용
binder/IServiceManager.h **class IServiceManager : public IInterface**	servicemanager 로 접근할 때 사용하는 접속 클래스


servicemanager
Gadd service
libbinder.so
User space
Linux Kernel
Binder Driver
드라이버 내의 서비스 노드
Node 0
.....
Node n


Application
JAVA
Native Service
GaddService
③
Client
transaction
①
service
registration
②
request
service
handle
Binder
Binder
IServiceManager
UserSpace
Kernel
KernelSpace
Binder Driver
Service
Manager
aesop.
Binder01

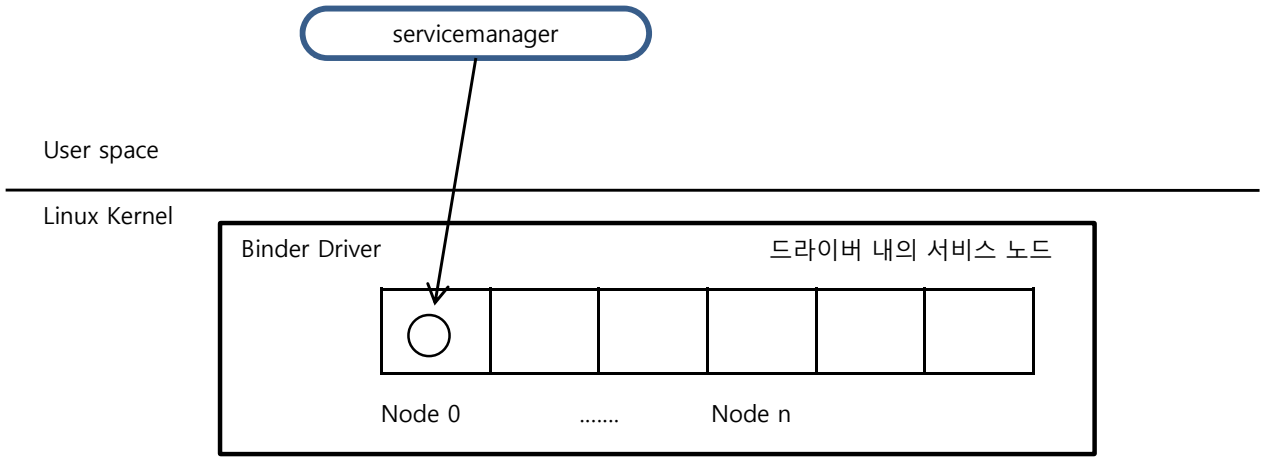

Native Service
AudioFlinger Service
AudioSystem Client
③
transaction
①
service registration
②
request service handle
Binder
Binder
libbinder
ServiceManager
UserSpace
KernelSpace
Kernel
Binderdriver
ashmem
Service Manager
media.audio_flinger
Memory sharing


도형표시:
상속
호출
Applications
MP3 Player
Applications Framework
MediaPlayer
Binder Client
libmedia.so
AudioTrack
BpAudioFlinger
BpInterface
libbinder.so
IPC
IAudioFlinger
IInterface
BnInterface
Binder Server
mediaserver
BnAudioFlinger
servicemanager
AudioFlinger
서비스 등록
Linux Kernel
Audio Driver
Binder Driver

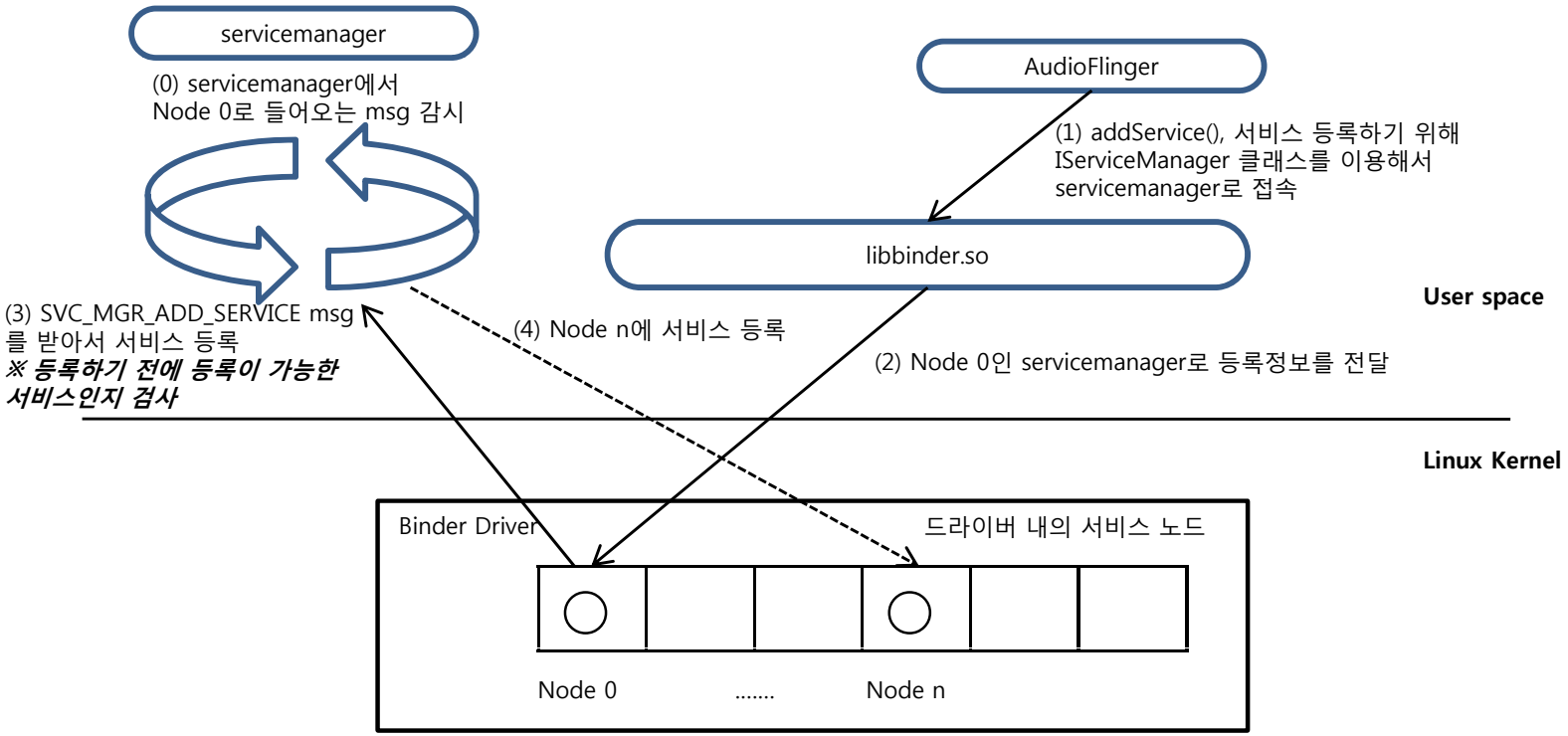

ICalc.cpp
ICalc
BpInterface
<ICalc>
BpCalc
IBinder
BBinder
BnInterface
<ICalc>
CalcService

- Binder 설명그림
  - http://www.aesop.or.kr/?mid=Board_Documents_AndroidPlatform&document_srl=431351

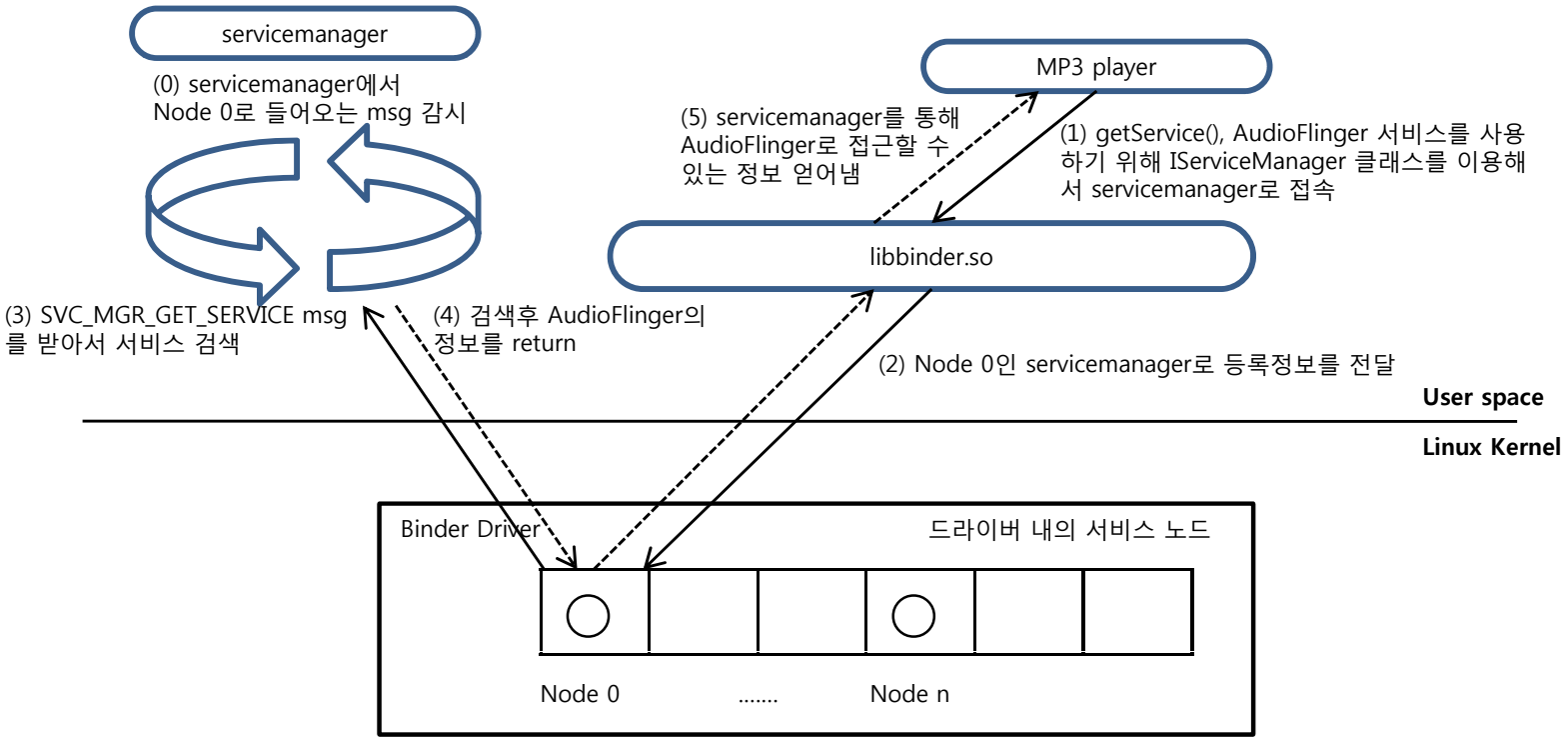
Service Manager 그림 1 – servicemanager 실행, Node 0에 등록



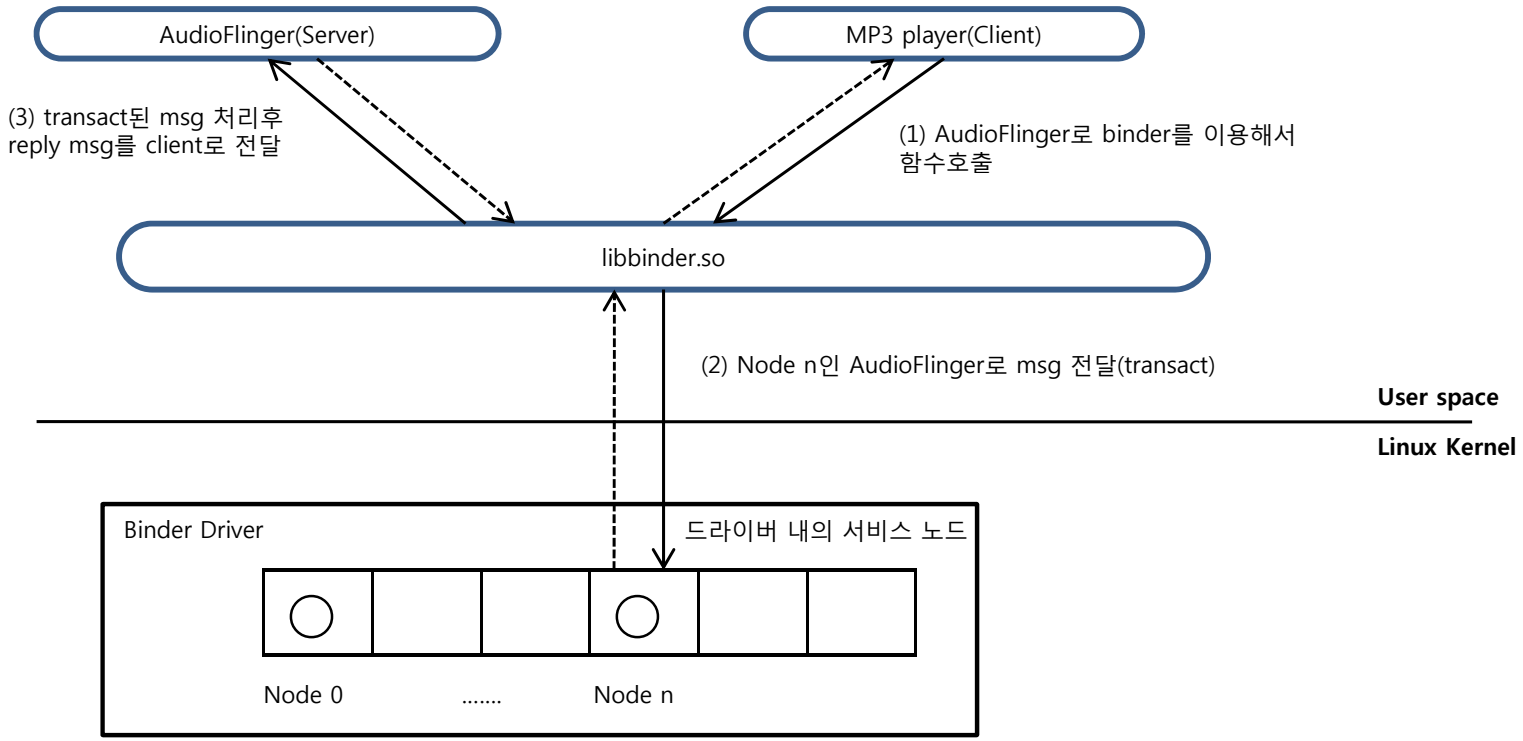
Service Manager 그림 2 – AudioFlinger 등록



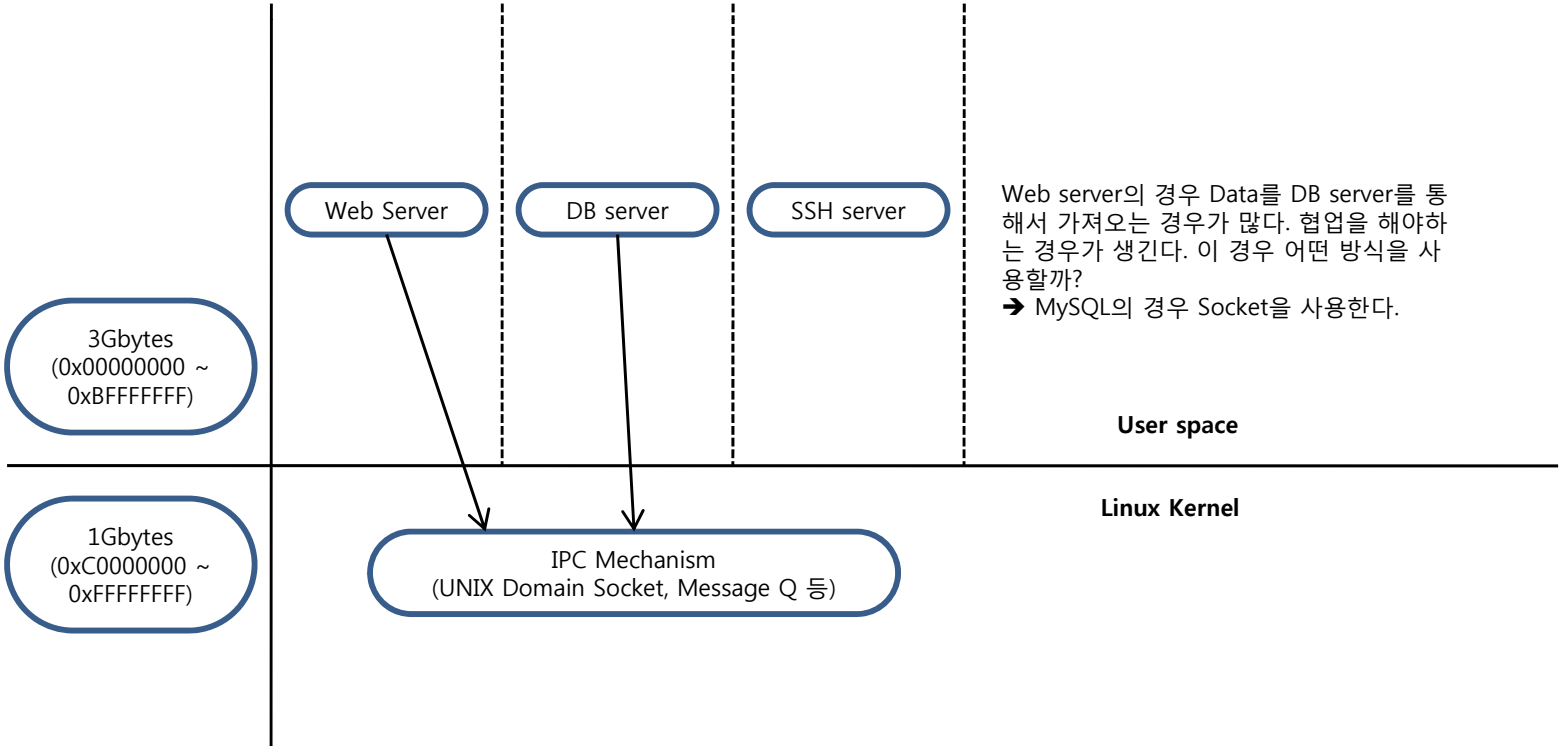
Service Manager 그림 3 – Client에서 AudioFlinger로 접속하기 위하여 servicemanager로 접속



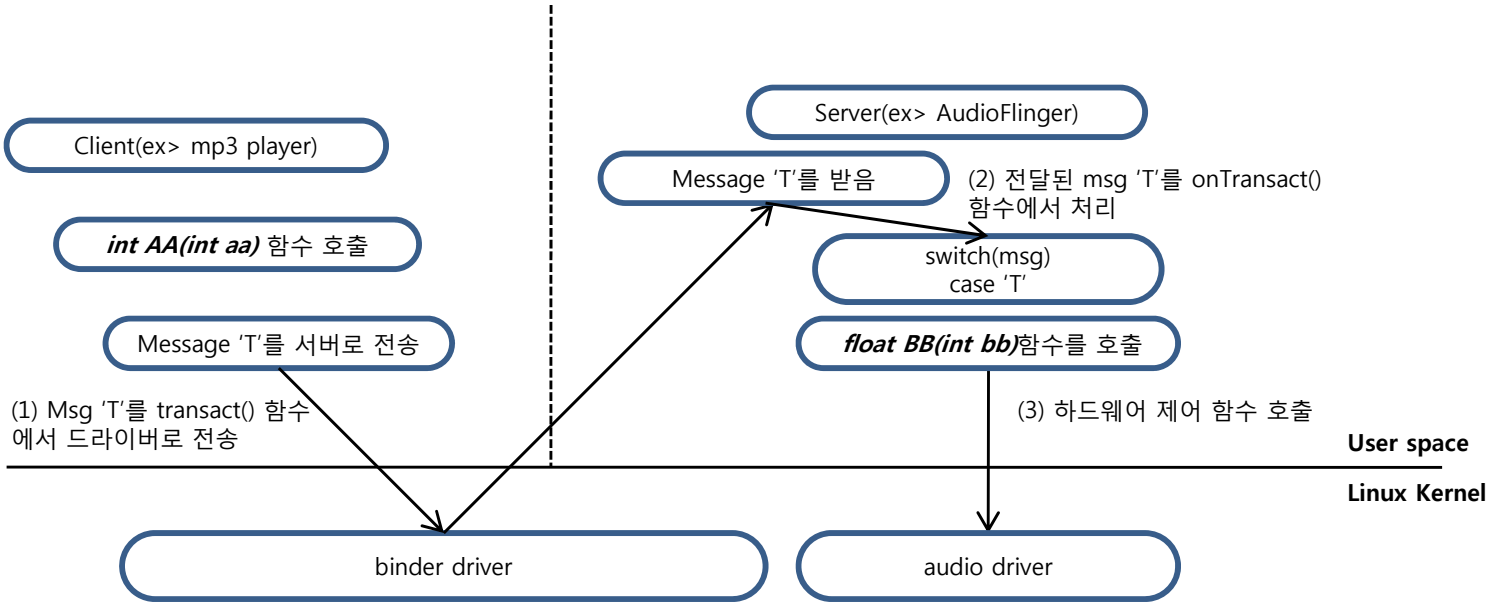
Service Manager 그림 4 – Client에서 AudioFlinger로 접속



IPC 필요성

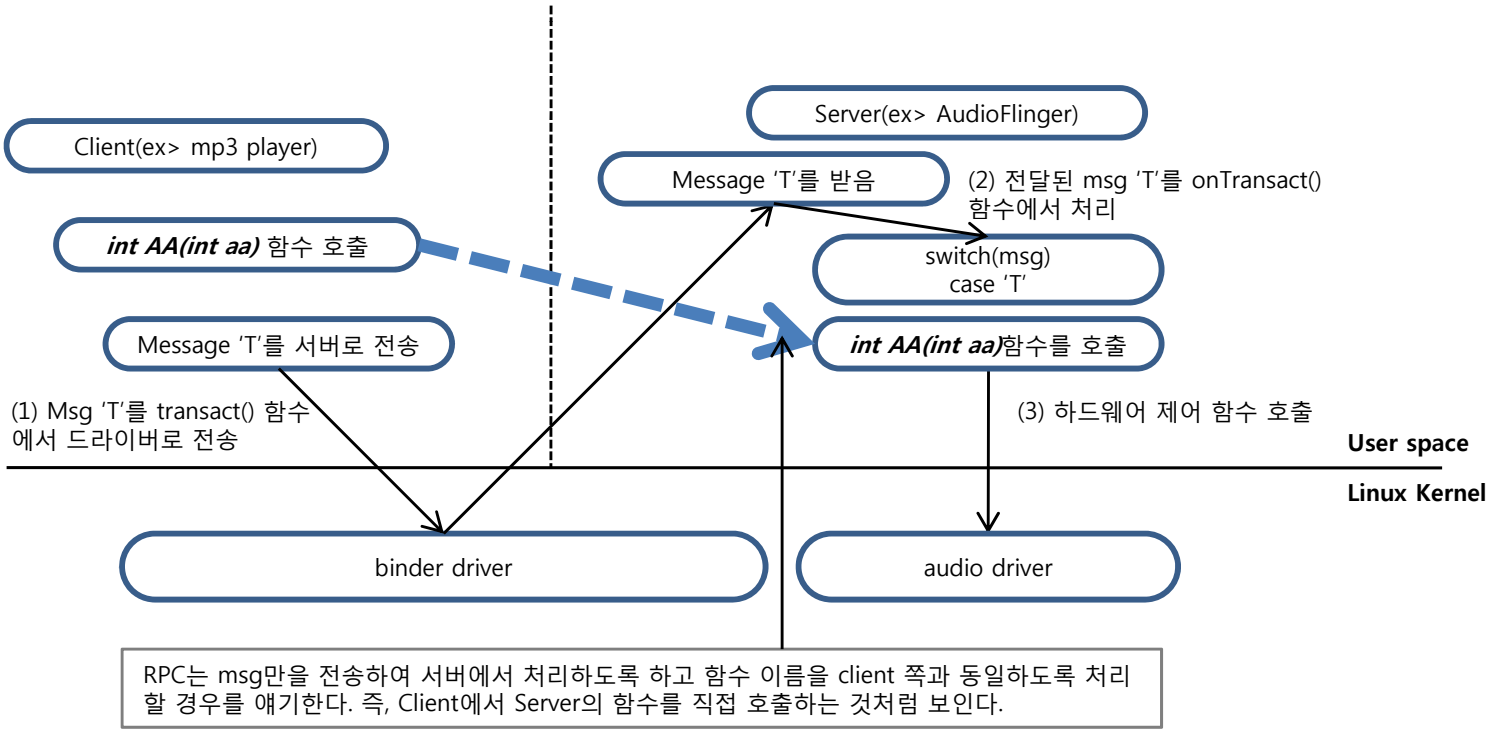

Web Server
DB server
SSH server
Web server의 경우 Data를 DB server를 통해서 가져오는 경우가 많다. 협업을 해야하는 경우가 생긴다. 이 경우 어떤 방식을 사용할까?
➔ MySQL의 경우 Socket을 사용한다.
3Gbytes
(0x00000000 ~
0xBFFFFFFF)
User space
Linux Kernel
1Gbytes
(0xC0000000 ~
0xFFFFFFFF)
IPC Mechanism
(UNIX Domain Socket, Message Q 등)

IPC 그림 - 간단하게 설명한 IPC 개념도

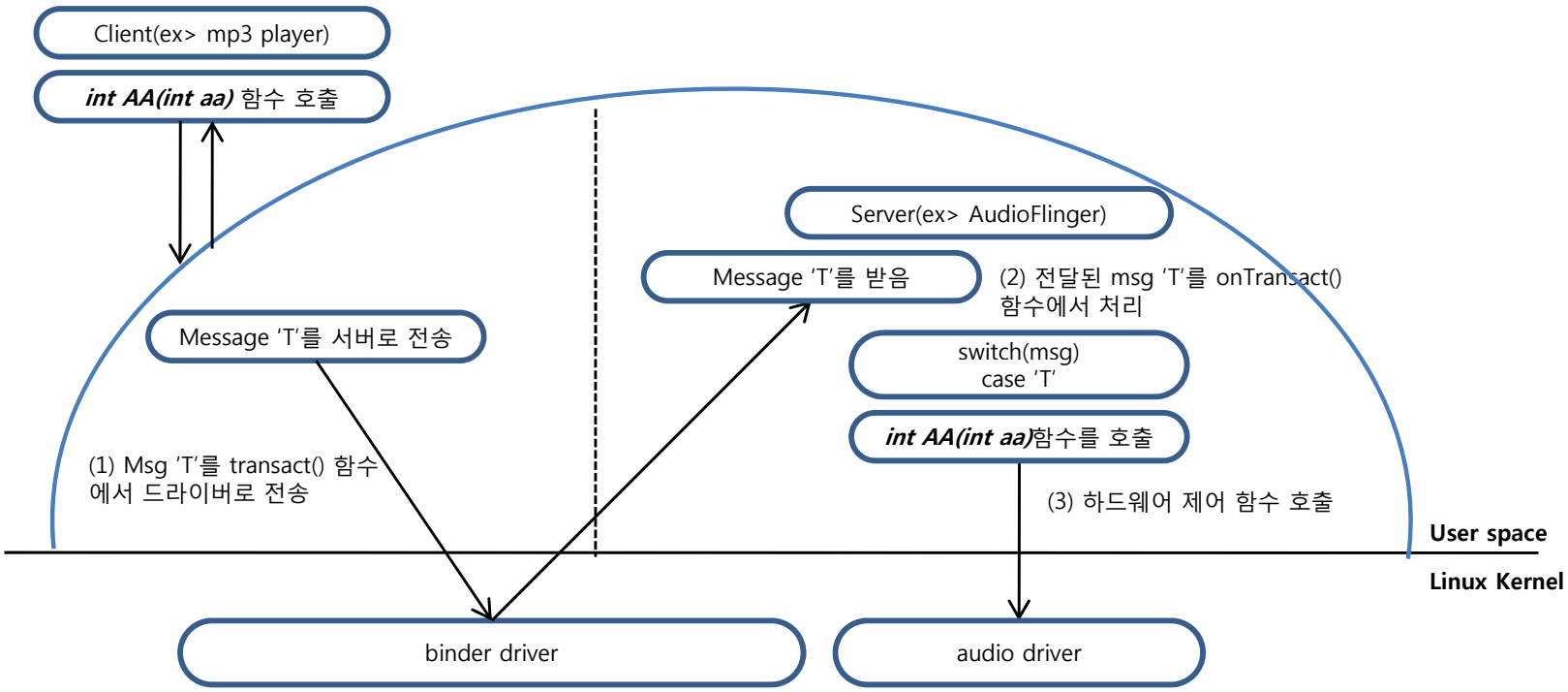


IPC는 msg만을 전송하여 서버에서 처리하도록 하되 함수 이름은 client 쪽과 동일하지 않도록 처리할 경우를 얘기한다. 즉, msg만 받아서 처리하는 구조이다.

RPC 그림 1 – 기본동작

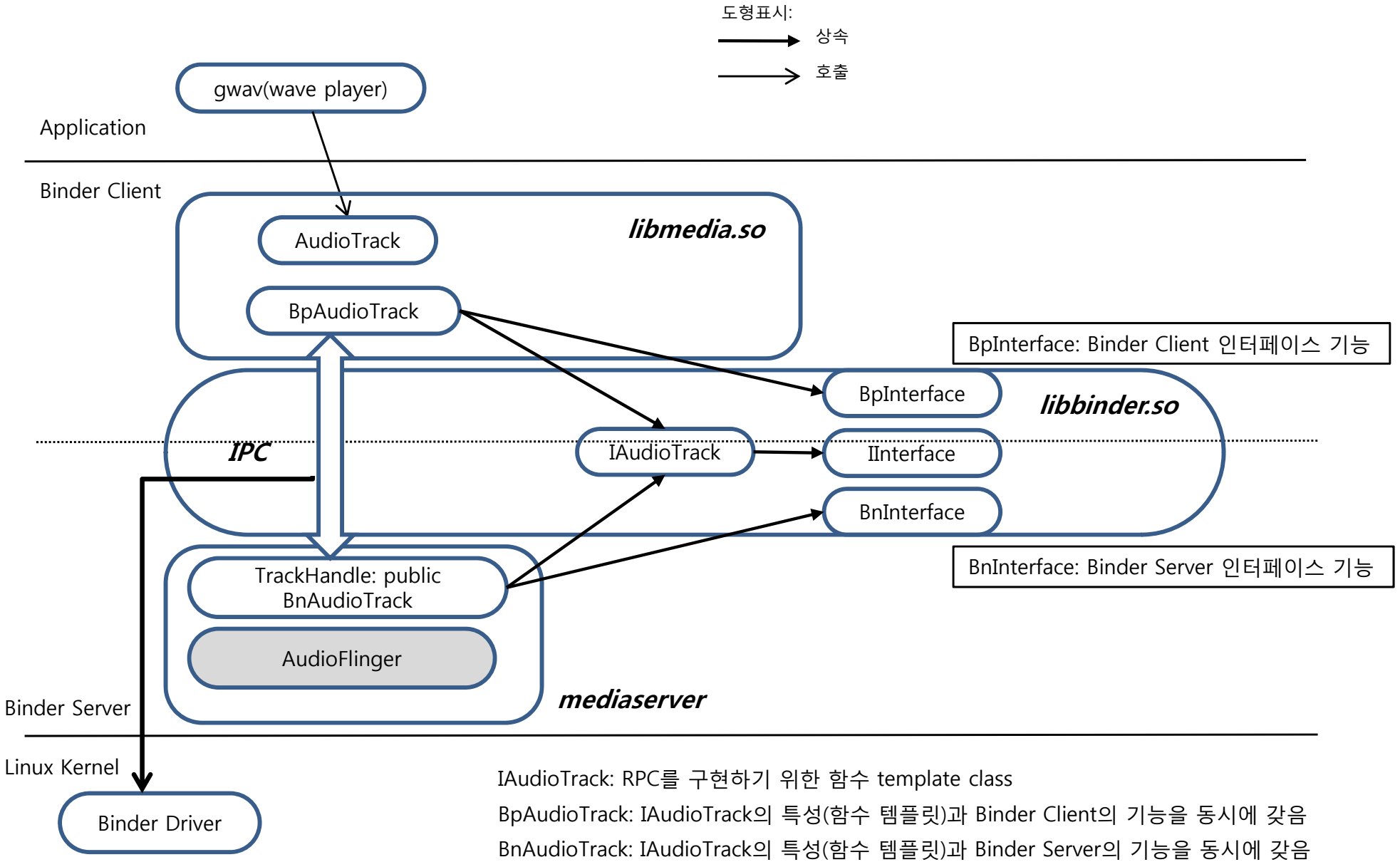

Client(ex> mp3 player)
Server(ex> AudioFlinger)
Message 'T'를 받음
(2) 전달된 msg 'T'를 onTransact() 함수에서 처리
int AA(int aa) 함수 호출
switch(msg)
case 'T'
Message 'T'를 서버로 전송
int AA(int aa)함수를 호출
(1) Msg 'T'를 transact() 함수에서 드라이버로 전송
(3) 하드웨어 제어 함수 호출
User space
Linux Kernel
binder driver
audio driver
RPC는 msg만을 전송하여 서버에서 처리하도록 하고 함수 이름을 client 쪽과 동일하도록 처리할 경우를 얘기한다. 즉, Client에서 Server의 함수를 직접 호출하는 것처럼 보인다.

RPC 그림 2 – 겉으로 보이는 클라이언트의 함수호출 동작



RPC는 IPC 통신 처리 부분을 client쪽에 보이지 않기 위하여 class화를 시켰다. 클라이언트 쪽에서는 Binder 통신 부분과 하드웨어 제어 부분 등이 보이지 않고, 그냥 일반 함수를 호출한 것과 같다.
※ Binder RPC는 응용 프로그램 작성자들이 편안하게 함수만 호출할 수 있도록 작성이 된 메커니즘이다.

PRC 그림 3 – 실제 RPC 구현 방법



IAudioTrack: RPC를 구현하기 위한 함수 template class

BpAudioTrack: IAudioTrack의 특성(함수 템플릿)과 Binder Client의 기능을 동시에 갖음

BnAudioTrack: IAudioTrack의 특성(함수 템플릿)과 Binder Server의 기능을 동시에 갖음

RRC 그림 4 – Binder를 알아야 하는 경우