공개SW 역량프라자

www.oss.kr

Android is not OS. It's Platform!

2011-05-19

유 명 환 (뻔뻔강사)

✉ funfun.yoo@gmail.com


강사 소개

강사 소개

(뻔뻔강사) 유 명 환

✉ funfun.yoo@gmail.com

@funfunyoo

- 충남대, 정보통신공학석사
- 한국전자통신연구원(ETRI) 및 해동정보통신 등 중소기업 근무
- **2004 : 대한민국 창업대전, 중소기업청장상 수상**
- **2005 ~ 2008 : ㈜넷플러그, 대표이사 역임**
- **현재, FUN-MARKET (www.fun-market.co.kr) 대표**
- 정보통신연구진흥원, 주간기술동향집필위원 선정 (2005)
- 정보통신연구진흥원, 과제평가위원 선정 (2005)
- 舊 정보통신부, IT 멘토 임명 (2007)
- 식품의약품안정청, 홈헬스케어 의료기기 전문위원 선정 (2009)
- 한국과학기술정보연구원, 과학기술정보협의회 위원 선정 (2009)
- 삼성전자, 삼성SDS, 삼성SDS멀티캠퍼스, LG전자, LG CNS,
  한국HP, 비트컴퓨터 등 다수 기관 교육 진행
- **한국 최초로 터널 내 무선 랜 기반 음성 방송 시스템 개발, 수락산 터널 외 전국 설치 (2006)**
- **한국 최초로 무선 USB 기반 플랫폼 개발 (2006), 2007 美 CES 전시회에서 공개됨**
- **실시간 소변 성분 분석기를 이용한 u-Health 솔루션 및 서비스 개발 (2008 ~ 2009)**
- **현재, Wi-Fi/BlutTooth 기반 솔루션 및 안드로이드 TV 리모컨 개발 컨설팅 중**

강사 소개

OS & Linux

Java Power Tool
저자 John Ferguson Smart
역자 강수영, 고영태, 백성민, 조정식

O'Reilly Media / 888 페이지 예상
2009년 12월 출간 예정

JavaScript: the Missing Manual
저자 David Sawyer McFarland
역자 김태곤

O'Reilly Media / 560 페이지 예상
2009년 12월 출간 예정

Maven: the Definitive Guide
저자 Sonatype Company
역자 소프트웨어 인 라이프

O'Reilly Media / 480 페이지 예상
2010년 1월 출간 예정

SCJP Sun Certified Programmer for Java 6 Exam 310-065
저자 David McFarland
역자 심재철

McGraw-Hill / 870 페이지 예상
2010년 1월 출간 예정

jBoss in Action
저자 Javid Jamae, Peter Johnson
역자 심재철

Manning Publication / 500 페이지 예상
2010년 출간 예정

Wicket in Action
저자 Martijn Dashorst, Eelco Hillenius
역자 황재선

Manning Publication / 450 페이지 예상
2010년 출간 예정

저자 소개

유명환 funfun.yoo@gmail.com

제미(FUN)있다는 이유 하나만으로 어플리케이션 개발자에서 임베디드 개발자로 전향하여, 중소기업 프로젝트부터 국가연구소 프로젝트까지 다년간 임베디드 프로젝트를 진행한 베테랑이다.

아무 것도 모르는 상태에서 임베디드 개발자로 거듭나기까지 겪었던 수많은 경험들 때문에 그 누구보다 처음 임베디드 기술을 접하는 개발자들의 마음을 헤아릴 줄 알기에 그들에게 도움이 되고자 2005년부터 현재까지 삼성전자, 삼성SDS, LG전자, LG CNS, 한국HP, 현대중공업, 한국전자통신연구소(ETRI), 국방과학연구소(ADD), 한국소프트웨어진흥원(KIPA), 한국정보산업연합회(FKII), KAIST EMDEC, 삼성SDS 멀티캠퍼스, 비트캠퍼스 등 다양한 기업 및 기관들에서 임베디드 관련 강의를 진행해오고 있으며, 2009년부터 "뻔뻔강사" 라는 별명으로 인터넷 카페 등에 임베디드 관련 강좌를 올리면서 수많은 개발자들의 호응을 받고 있다.

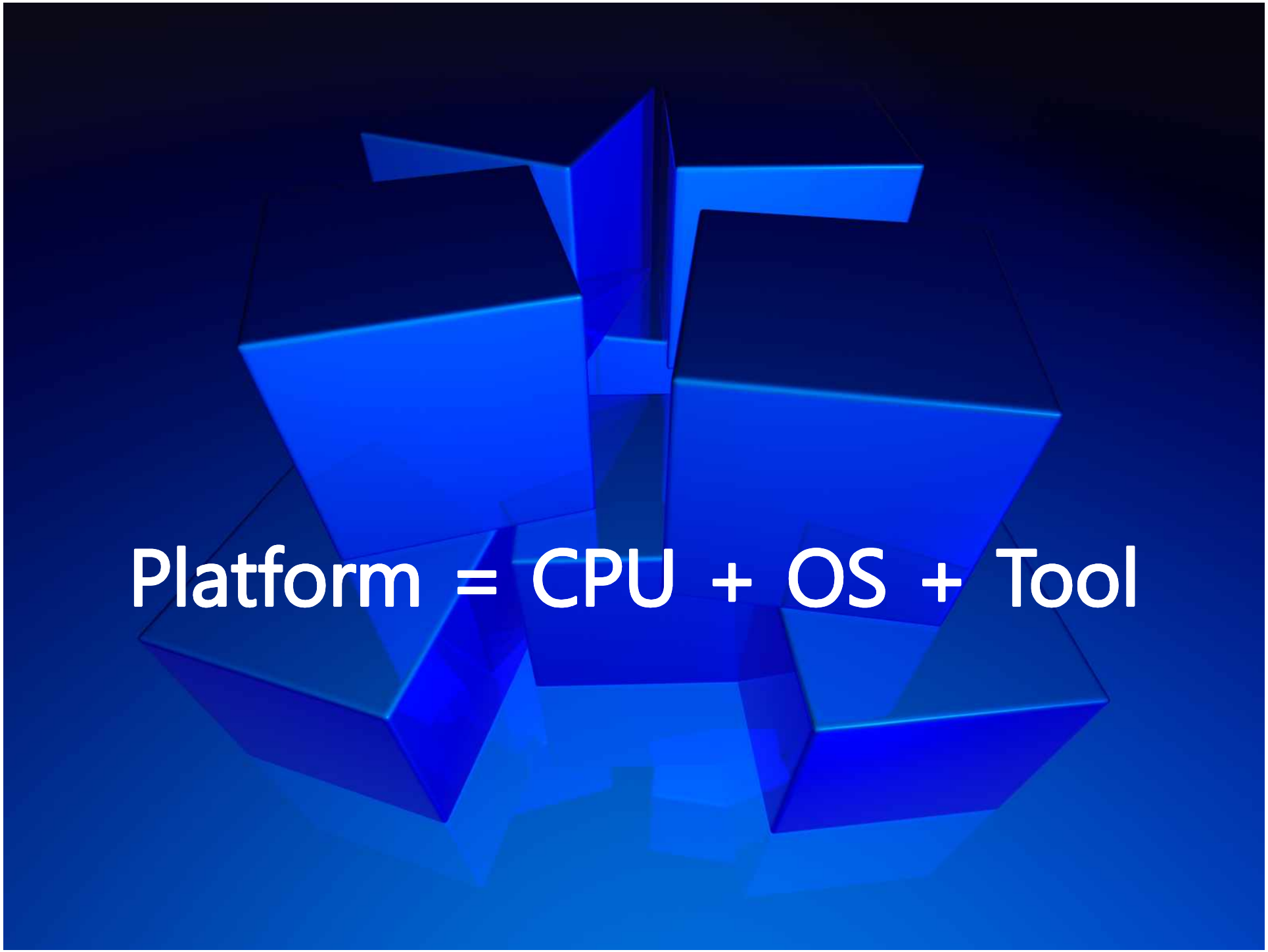
"FUN = Funny(재미있고) + Useful(유익하며) + New(새로운)" 지식 전파의 달인으로 거듭나기 위해 지었다는 뻔뻔강사라는 별명답게, 현재 인터넷 사이트 FUN-MARKET(www.fun-market.co.kr)을 통해 기존에 나왔던 교재들과는 사뭇 다른 교재와 강좌, 교육용 키트 등을 보급하고 있으며, 최근에는 로봇과 구글 안드로이드(Android)를 주제로 작술하고 있다.

jmson.tistory.com

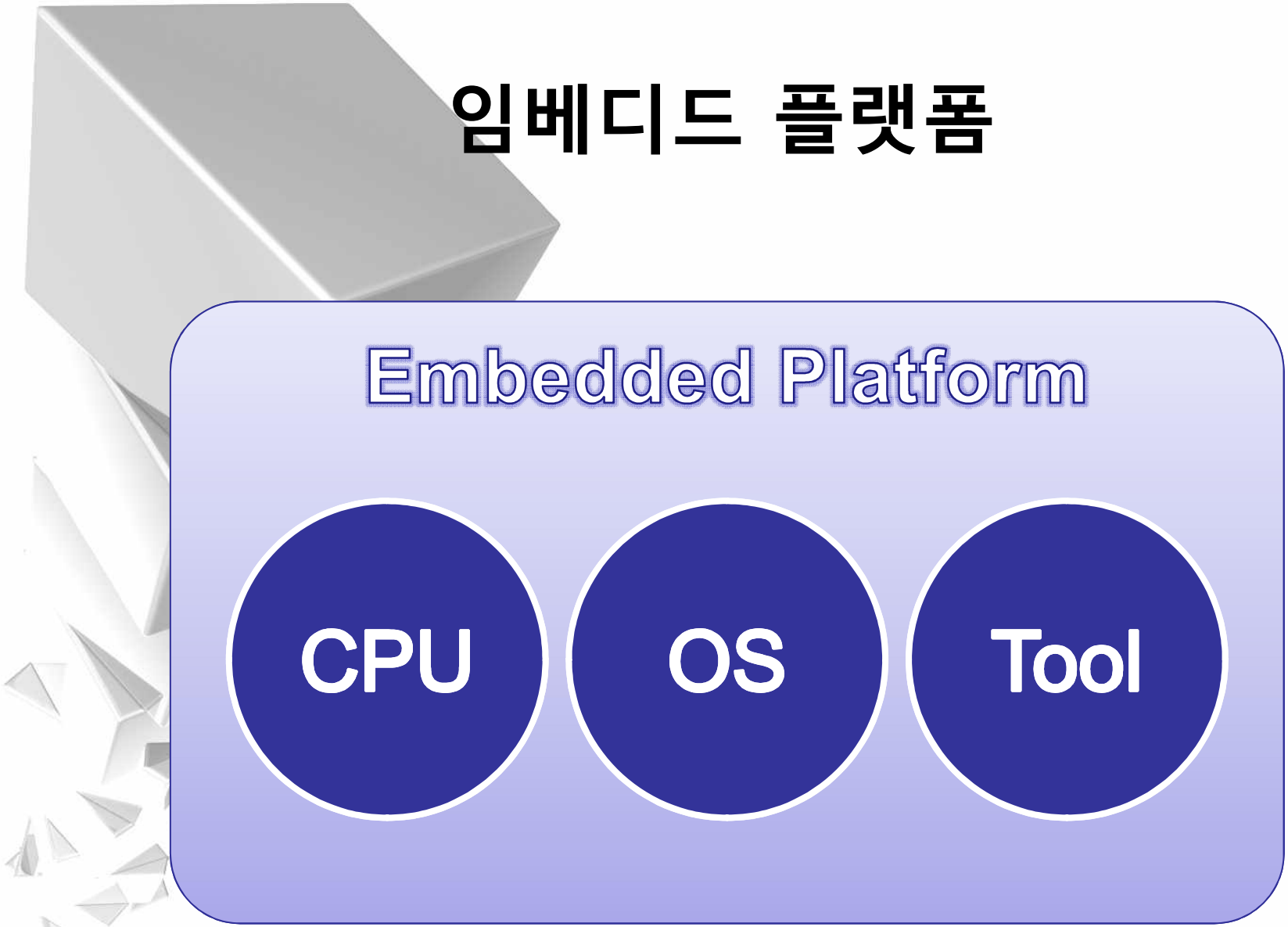
www.funfunstudy.co.kr 뻔뻔강사 교육 포털 사이트

Index

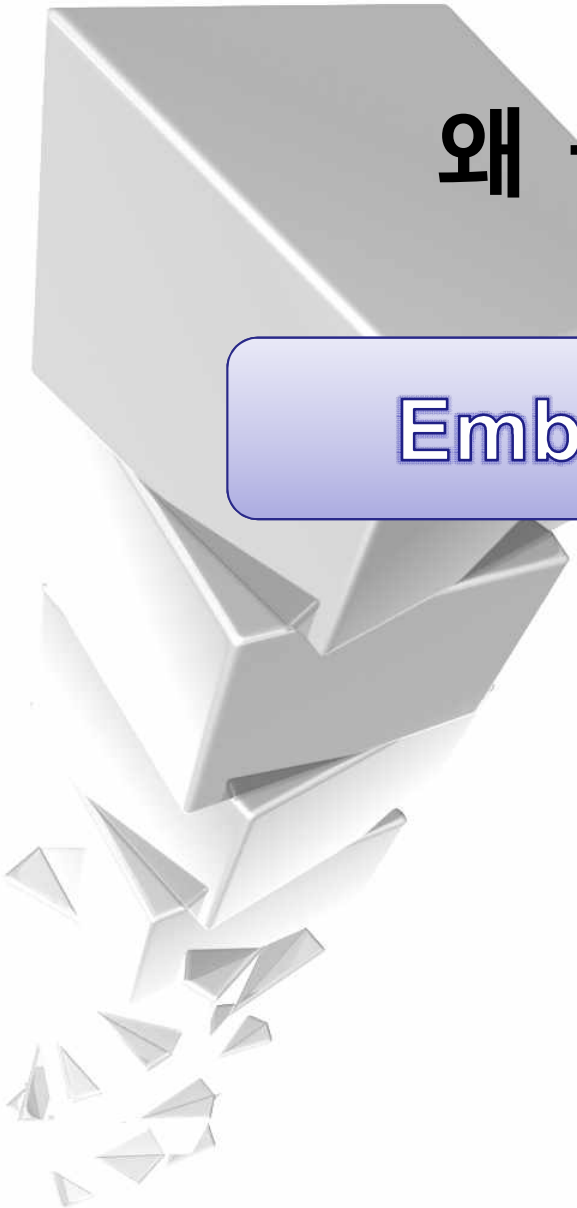
1. Platform = CPU + OS + Tool
2. 리눅스 계층별 내부 구조 분석
3. 안드로이드 vs 리눅스


Platform = CPU + OS + Tool

임베디드 플랫폼



안드로이드 플랫폼



왜 플랫폼이 중요한가?

Embedded = Cost Saving

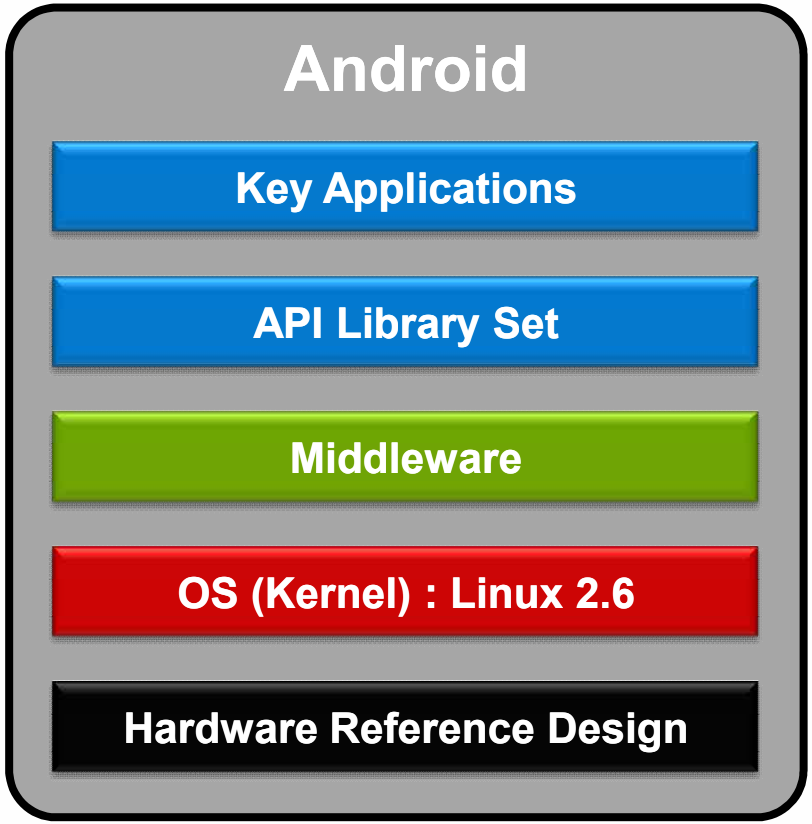
비용 = (눈에 보이는 비용) + (눈에 보이지 않는 비용)

눈에 보이는 비용 = 부품 구매비 + 생산비 + …

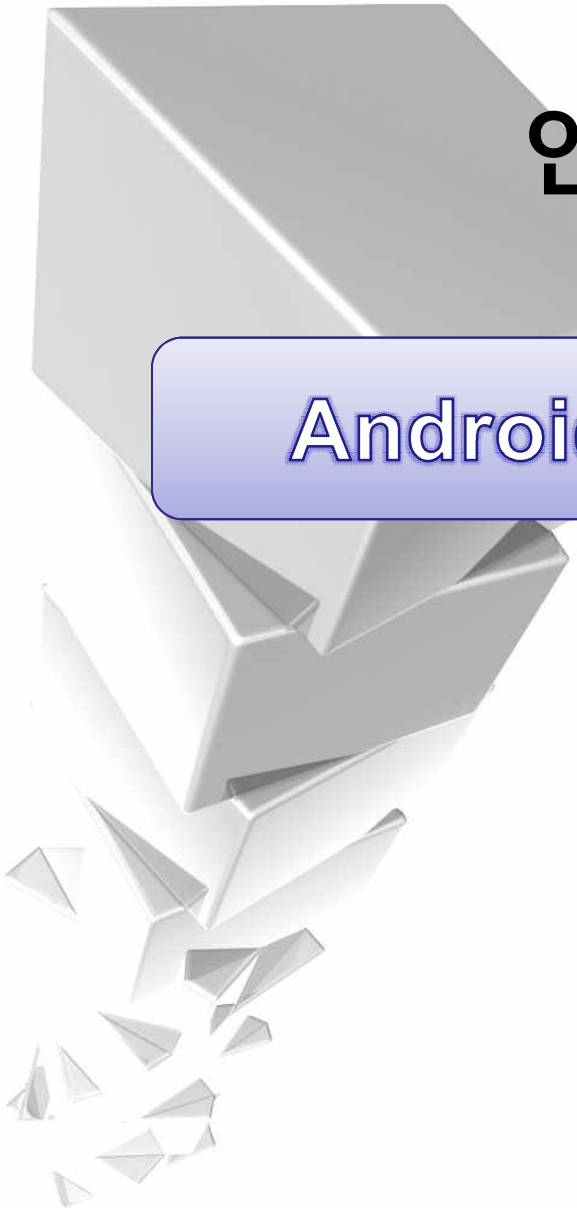
눈에 보이지 않는 비용 = 인건비 + …

안드로이드 플랫폼

스타워즈 : 모바일 플랫폼 전쟁



안드로이드 = 모바일 기기 개발을 위한 오픈 소스 플랫폼



안드로이드 플랫폼

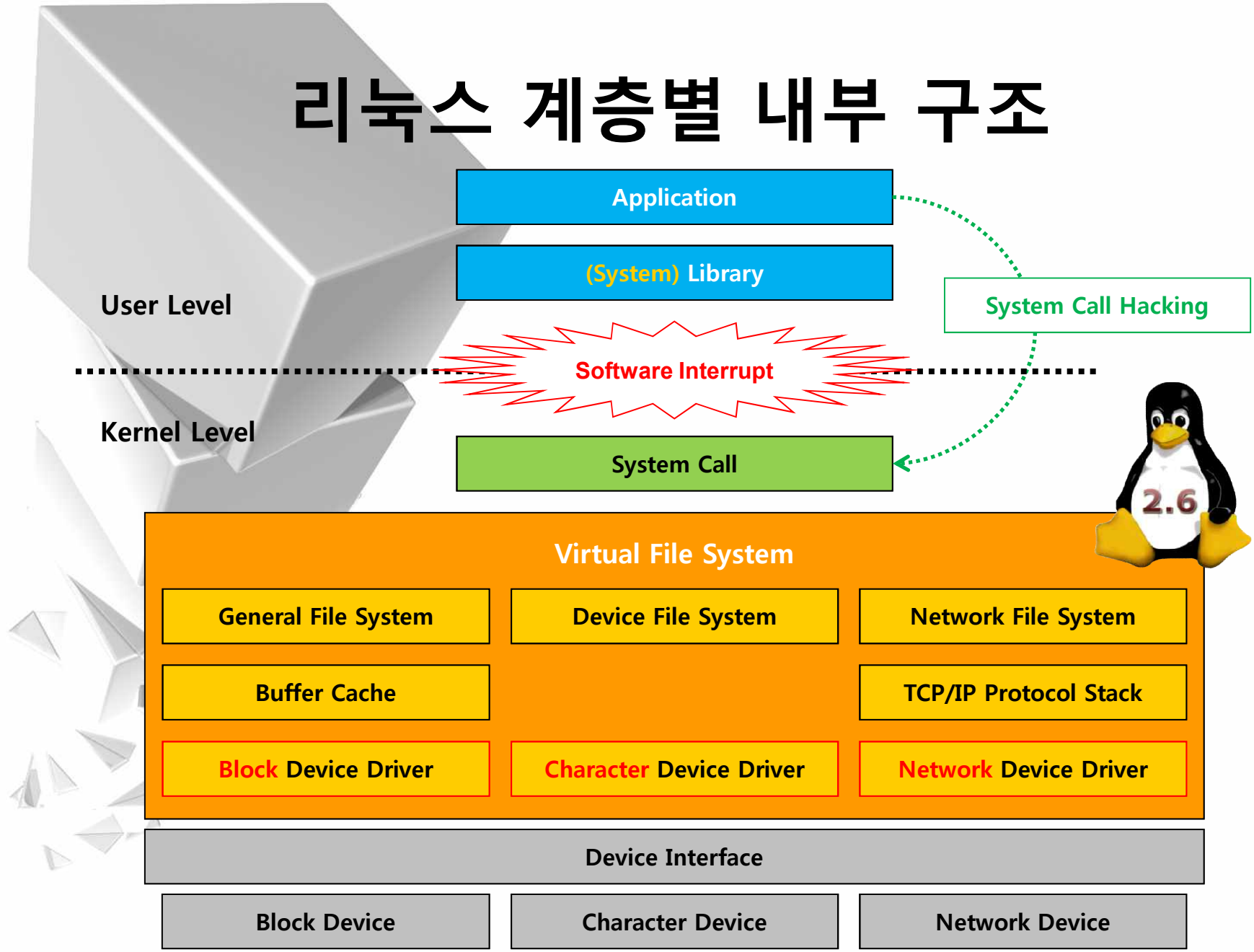
Android = XML + Java + Linux

XML, Java, Linux = 이미 잘 알려져 있고 많이 사용되고 있는 기술

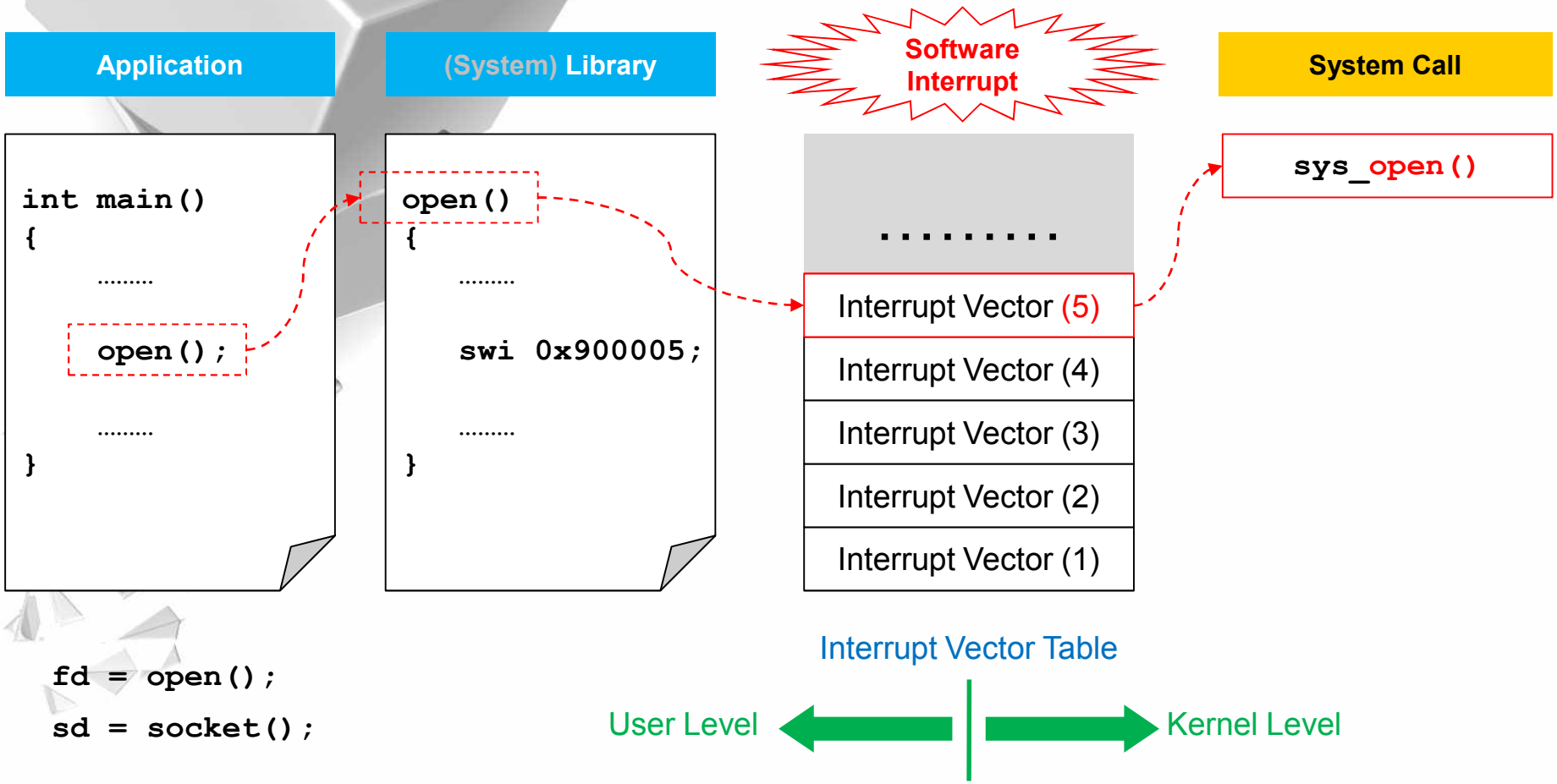
잠재된 개발자가 많다 -> 어플리케이션이 풍부해질 것이다 -> 플랫폼을 장악할 것이다


리눅스 계층별 내부 구조 분석

리눅스 계층별 내부 구조

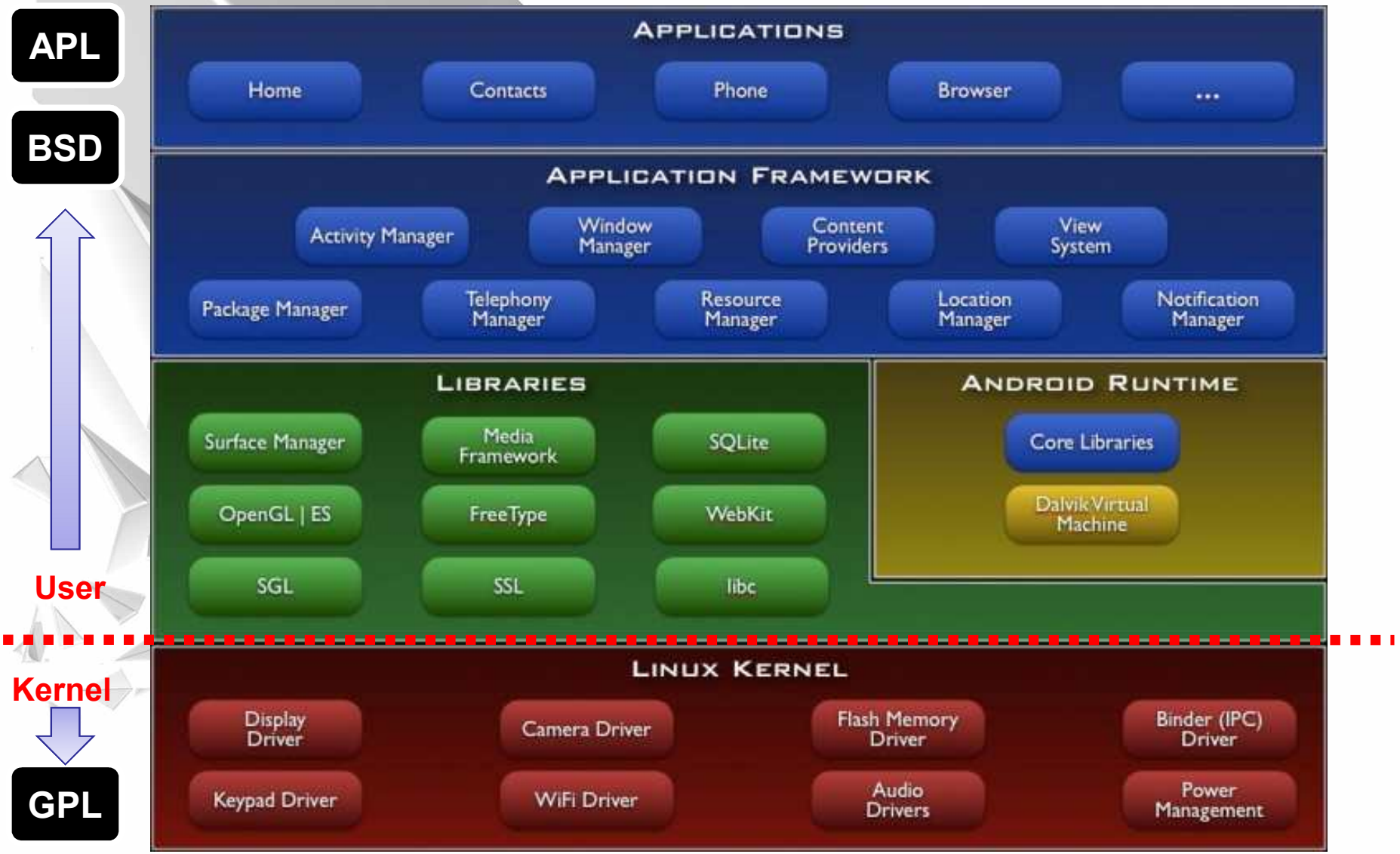

Application
(System) Library
User Level
System Call Hacking
Software Interrupt
Kernel Level
System Call
2.6
Virtual File System
General File System
Device File System
Network File System
Buffer Cache
TCP/IP Protocol Stack
Block Device Driver
Character Device Driver
Network Device Driver
Device Interface
Block Device
Character Device
Network Device

리눅스 시스템 콜 함수 호출 구조

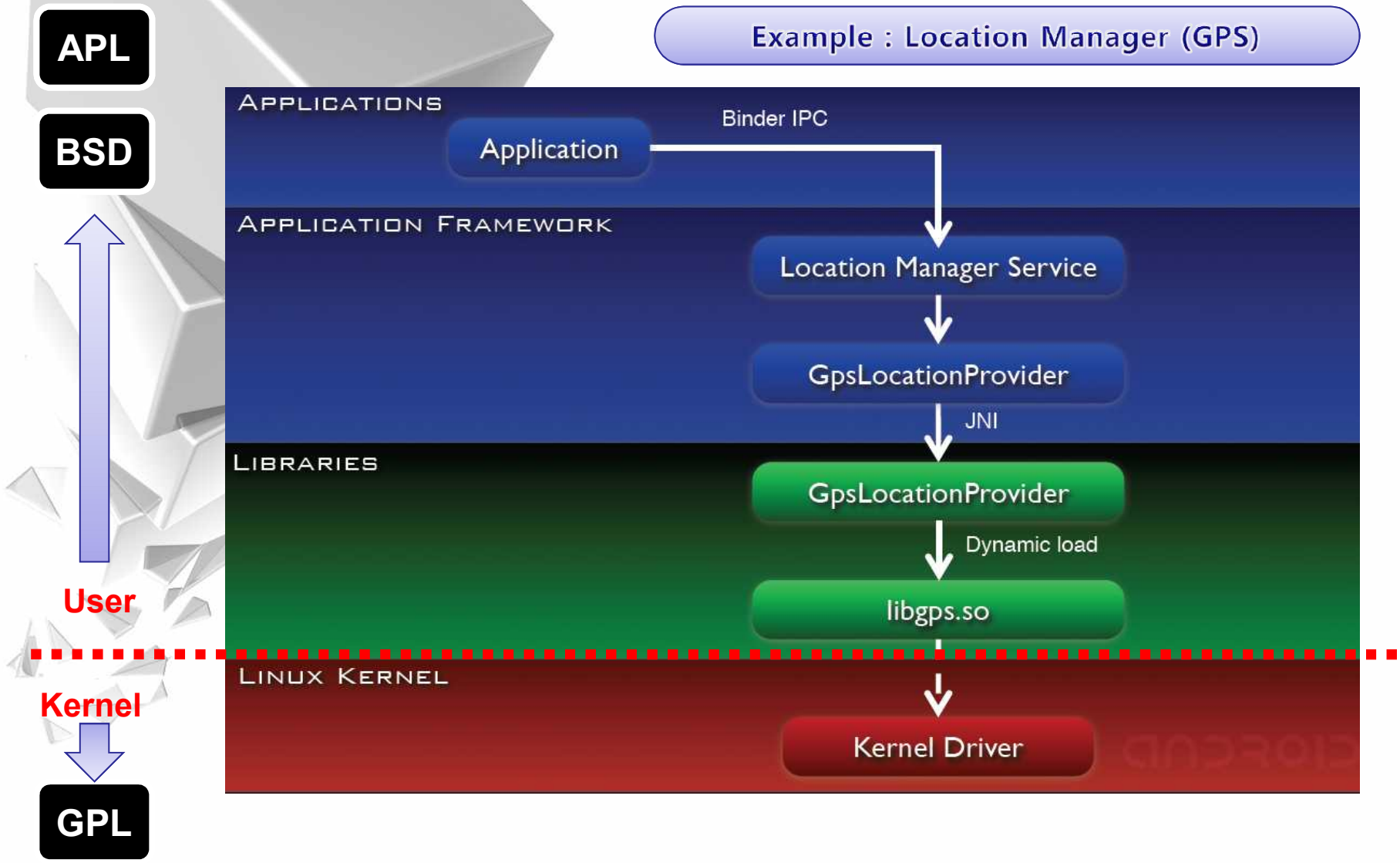



안드로이드 vs 리눅스

안드로이드 계층별 내부 구조



안드로이드 계층별 내부 구조 : 사례



HAL : Hardware Abstraction Layer

JNI 를 사용하는 이유

- 속도 문제
- Java 에서 하드웨어 제어
- Java 에서 지원되지 않는 특정 운영체제(OS) 서비스와 기능 연동

기존 리눅스와 안드로이드의 드라이버 차이점

- 기존 임베디드 리눅스는 장치 드라이버들이 커널 영역(Kernel Level)에서 동작
- 반면 안드로이드는 장치 드라이버들이 유저 영역(User Level)에서 동작

-> 커널 영역을 제외한 HAL 을 의미!!!

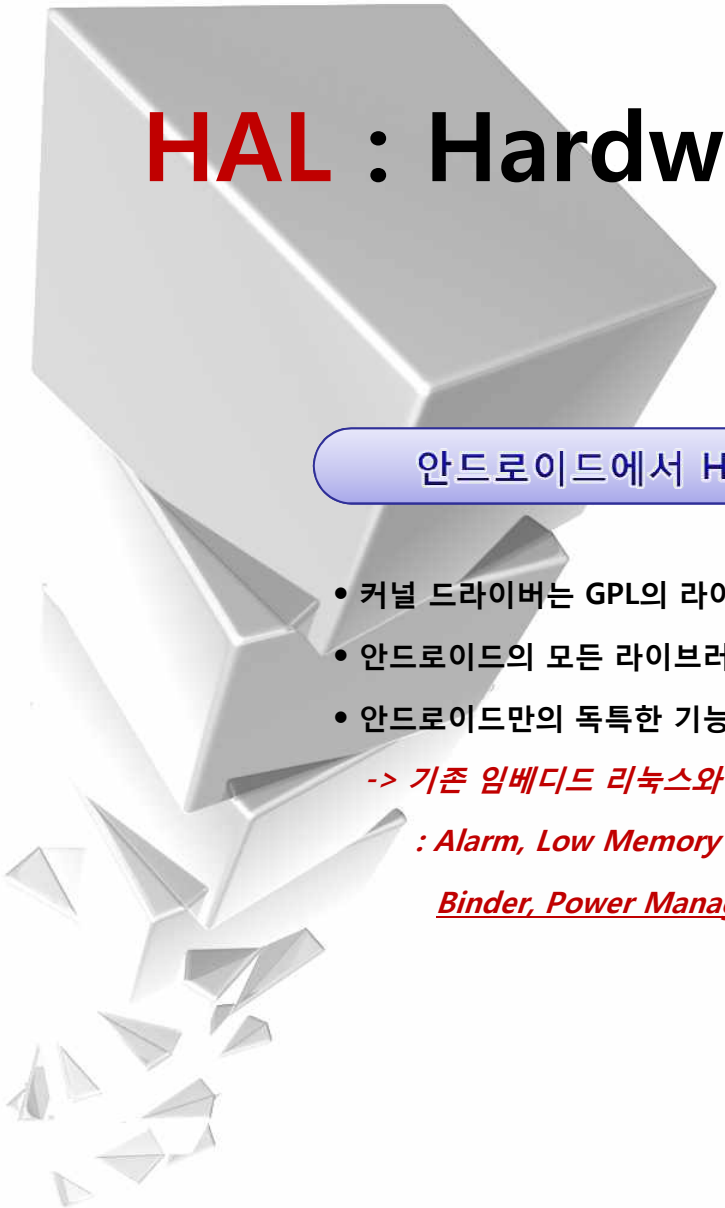
-> MMU 등을 참조하는 등의 추가적인 작업이 필요하여 조금 불편할 수 있지만,

자신에게 할당되어 있는 메모리 번지만 접근할 수 있기 때문에 보다 더 안전!!!

안드로이드에서 드라이버들을 유저 영역에 넣은 이유?

- 라이선스 문제
- 안정성
- 플러그인 방식으로 드라이버 배포 가능

HAL : Hardware Abstraction Layer



안드로이드에서 HAL 과 같은 미들웨어를 만든 이유?

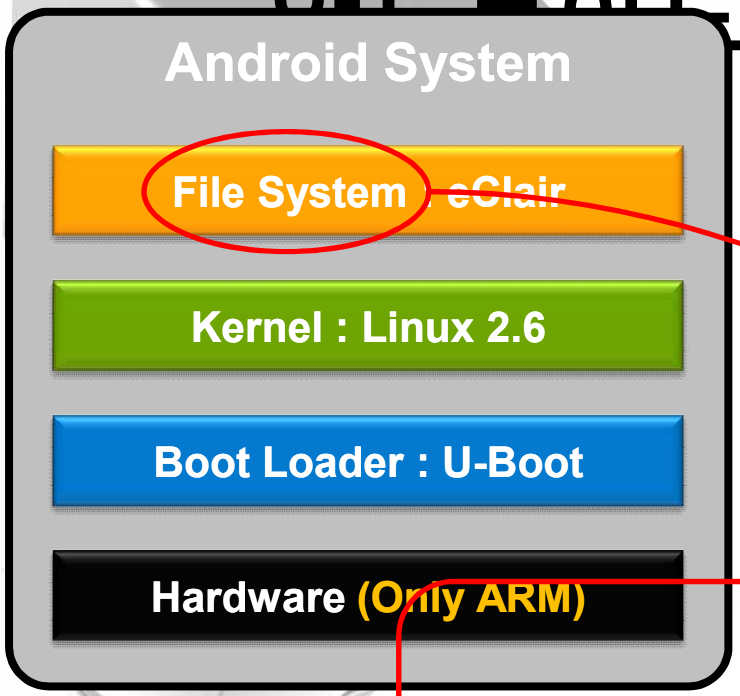
- 커널 드라이버는 GPL의 라이선스에 적용을 받는다는 문제
- 안드로이드의 모든 라이브러리들이 표준화되어 있지 않다는 문제
- 안드로이드만의 독특한 기능(Binder(IPC), Power Management 등)을 부여하기 위해

-> 기존 임베디드 리눅스와 달리 안드로이드 용 리눅스 커널에서 추가된 커널 구성 요소들

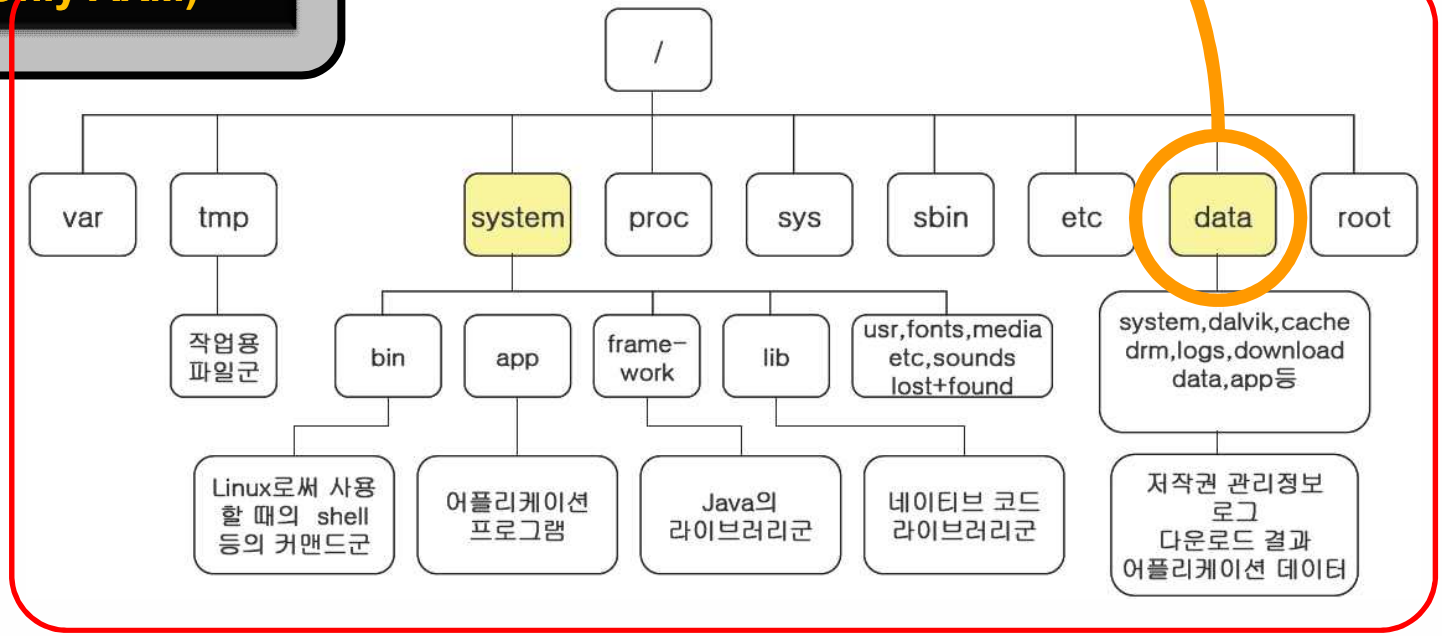
: Alarm, Low Memory Killer, Shared Memory Driver(Ashmem), Kernel Debugger,

Binder, Power Management, Logger

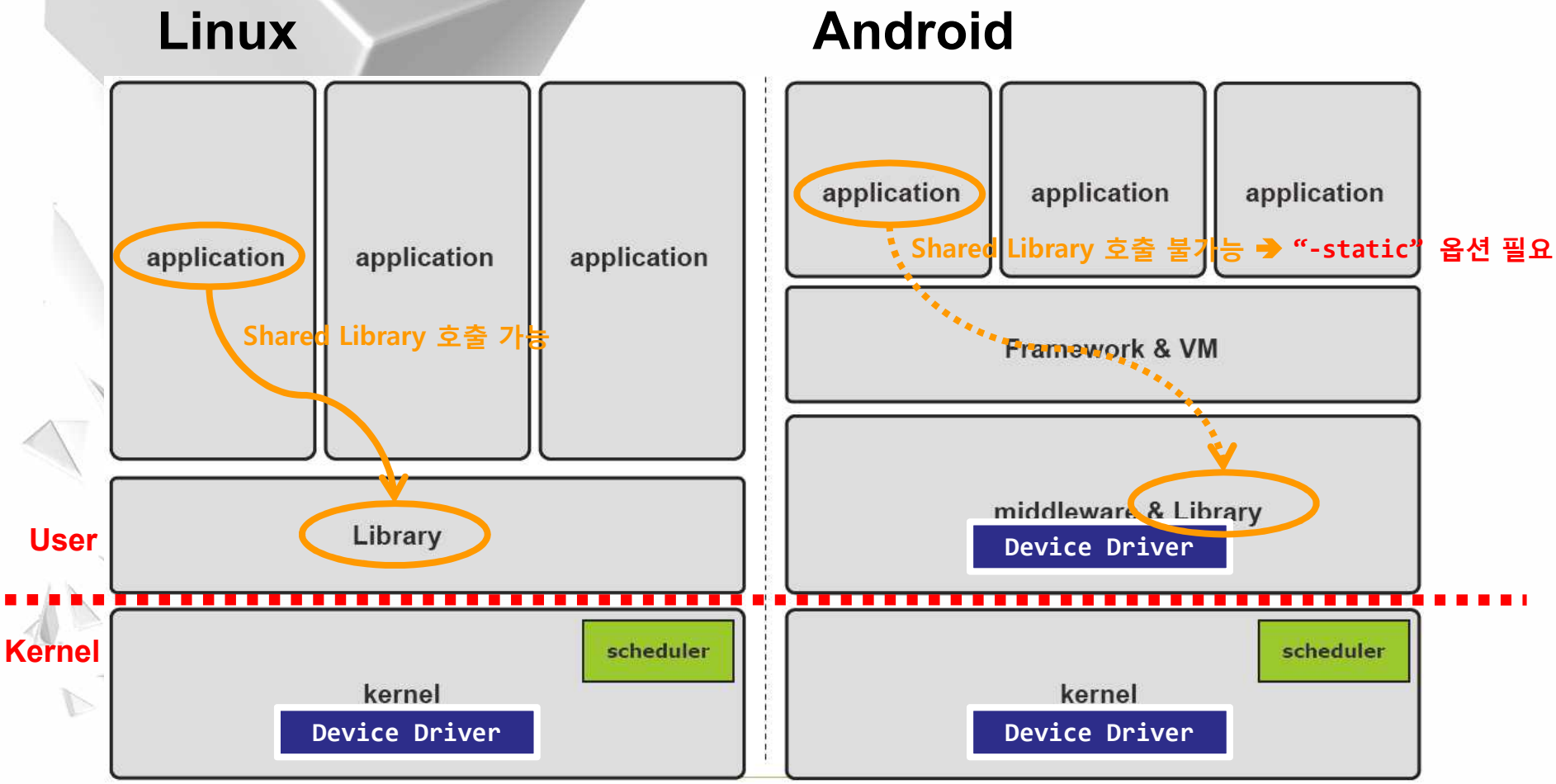
안드로이드 파일 시스템 구조



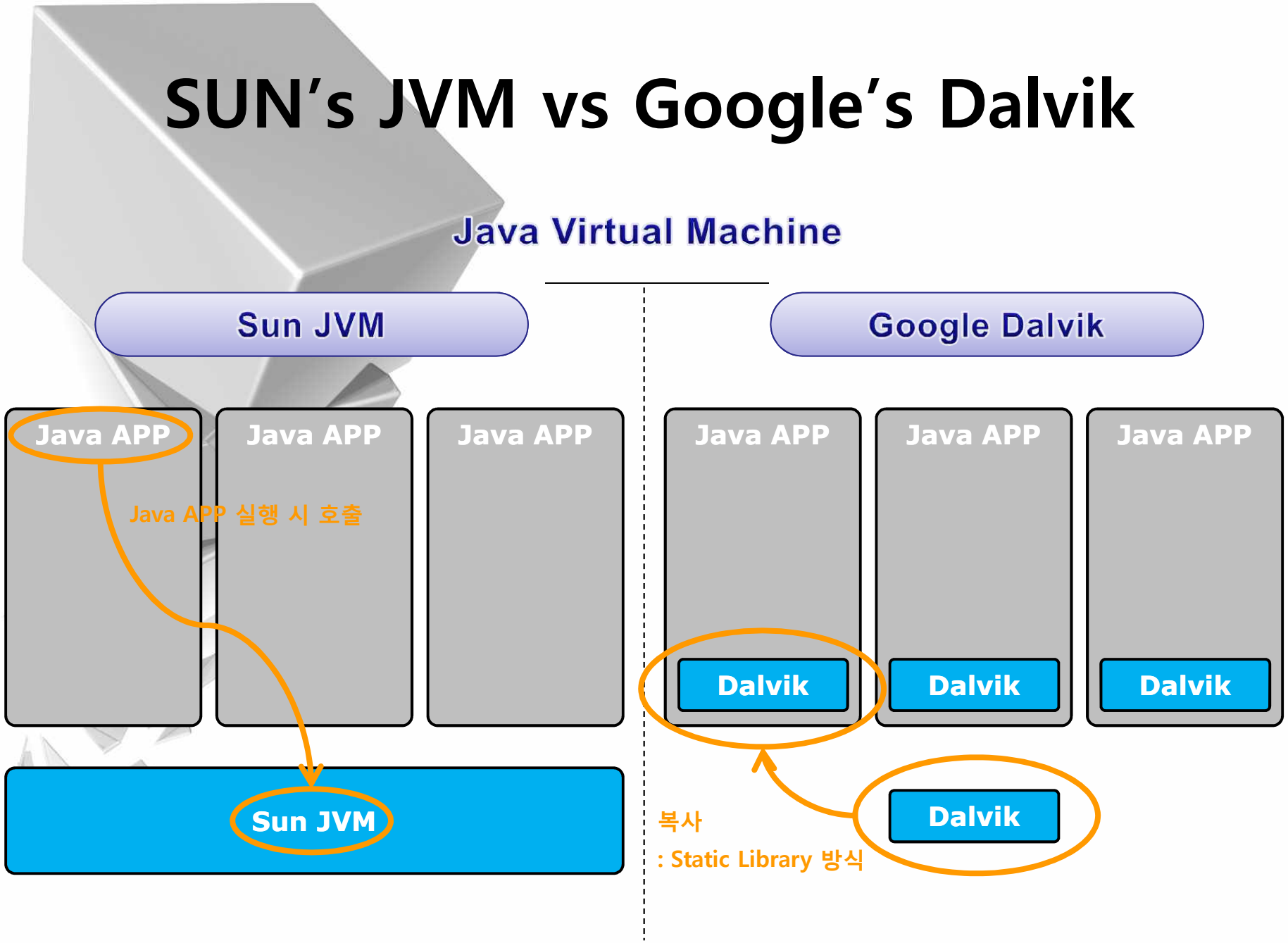
C/C++ 안드로이드 APP 은
단말기(혹은 Eclipse 에뮬레이터)의
/data 디렉토리에 복사하여 실행!!!



안드로이드 vs 리눅스 계층별 구조



SUN's JVM vs Google's Dalvik



안드로이드 앱에 main 이 없는 이유?

main 함수(메쏘드)가 없는 소스 코드는 없다!!!

Android Application Source

```
package com.funfun.Hello23v;

import android.app.Activity;
import android.os.Bundle;

public class Hello23v extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

RTOS Application Source

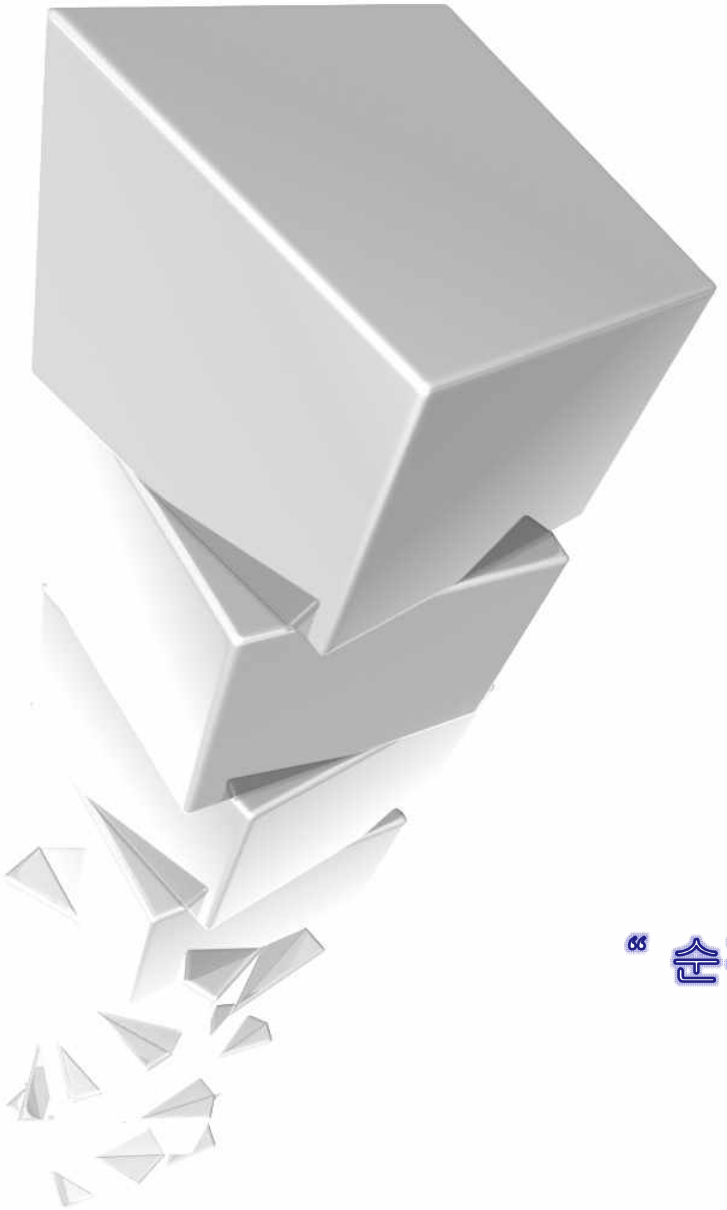
```
void task_start(void *data)
{
    ......
}

int main(void)
{
    OSInit();

    OSTaskCreate(task_start, (void *)0,
      (void *)&task_start_stk[OS_TASK_DEF_STK_SIZE - 1],
       0);

    OSStart();

    return 0;
}
```



Q & A

“ 순간의 □ □ □ 이 평생을 좌우한다! ”