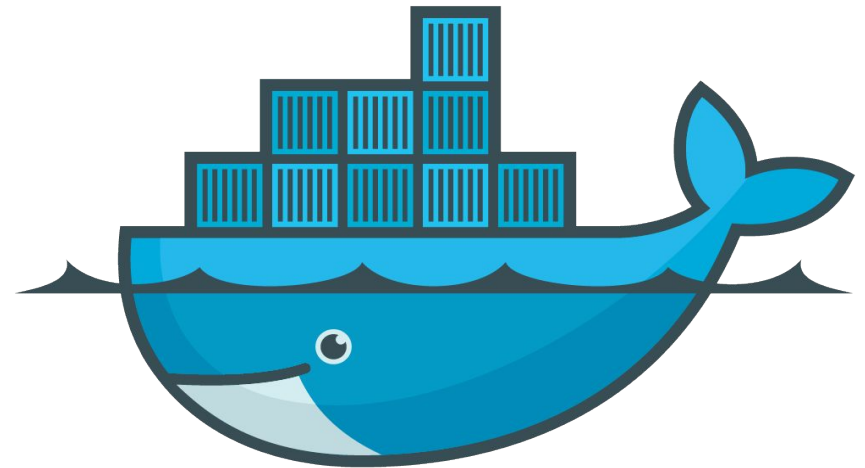
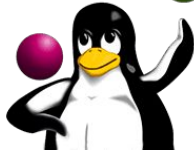




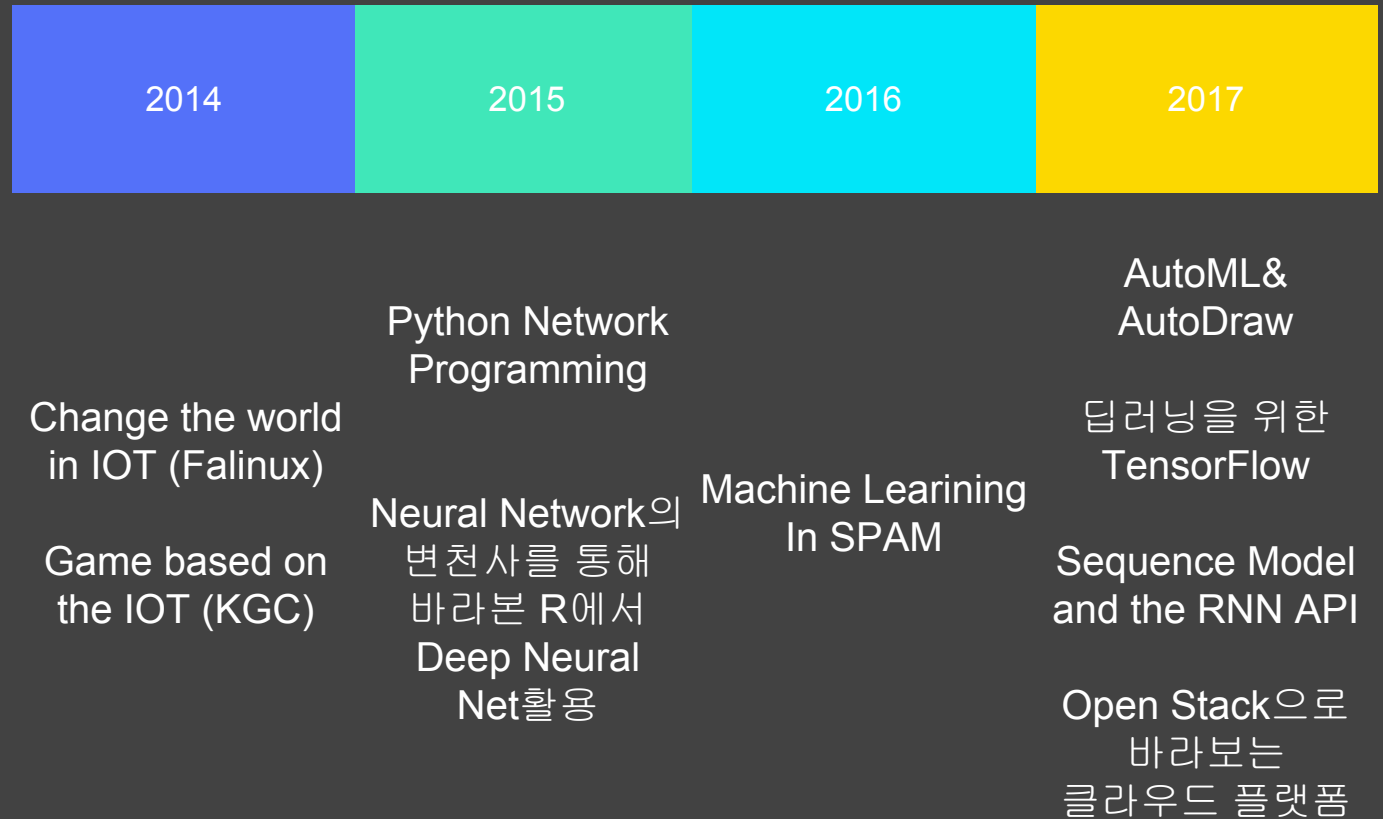
KVM



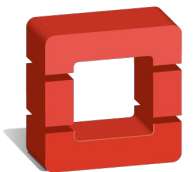
docker

Produced by Tae Young Lee

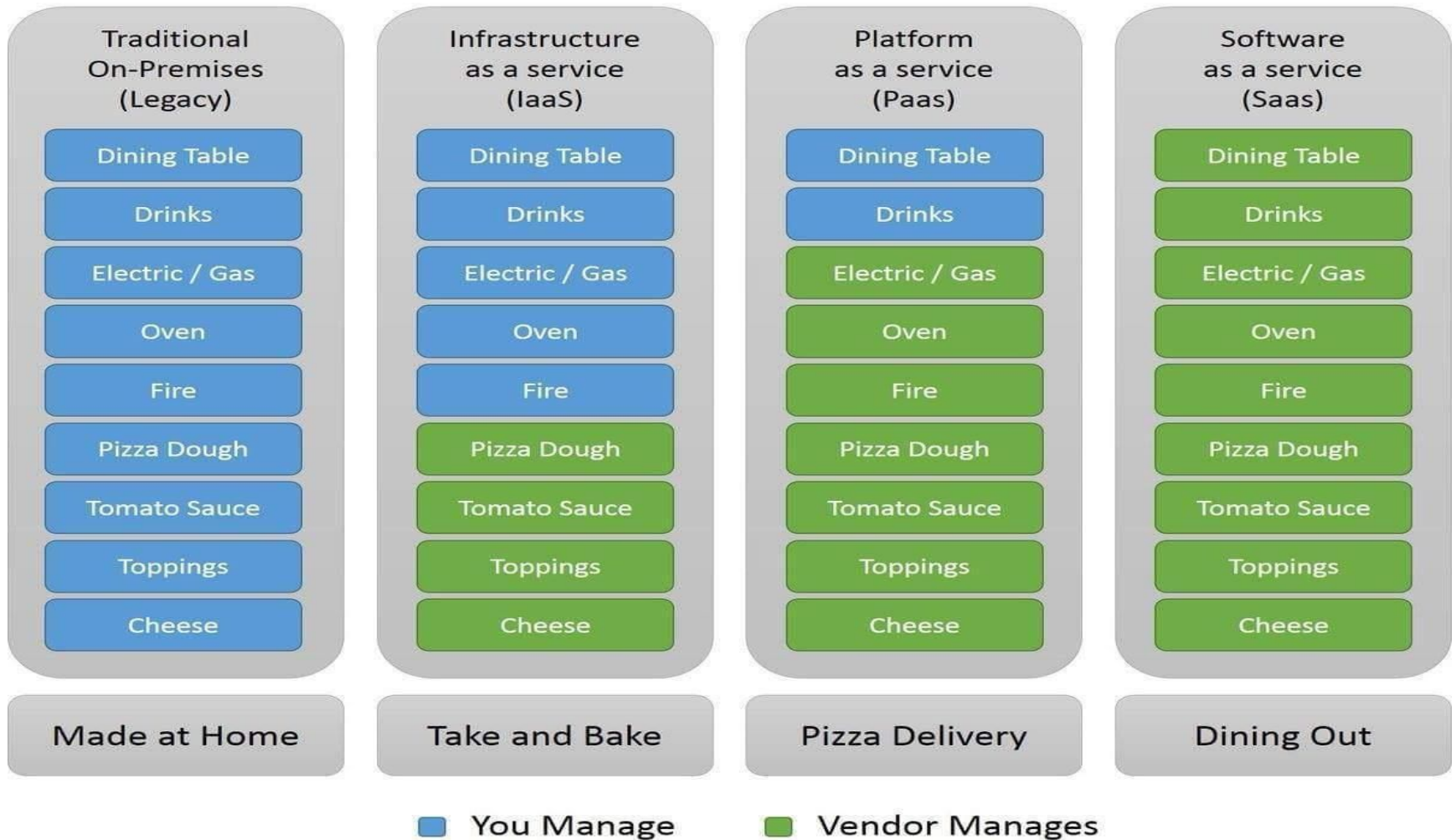


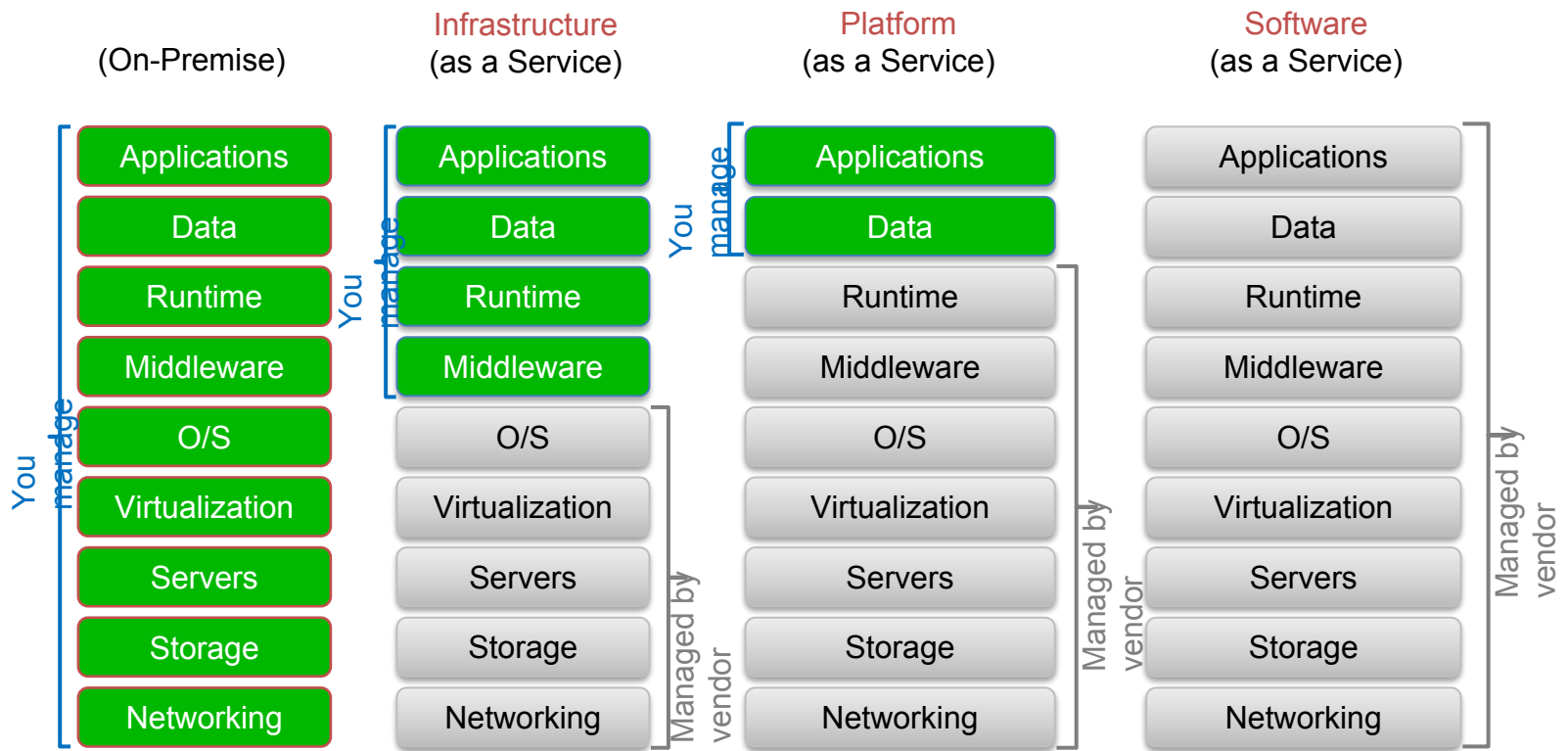


가상화 개념 잡기



Pizza as a Service





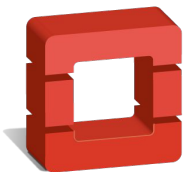
VS



openstack™

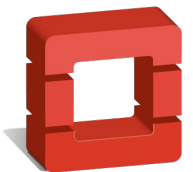
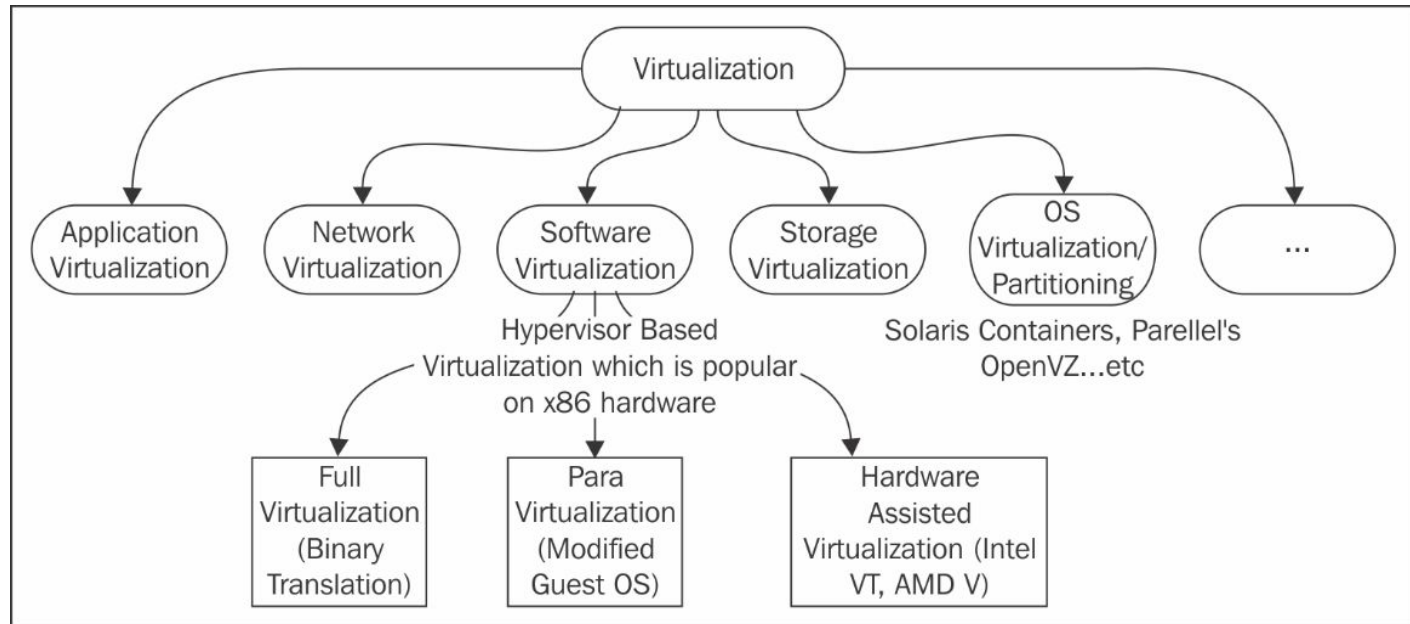
가상화의 장점

- 서버통합
 - 데이터센터 전력 에너지와 공간 절약
- 서비스 분리
 - 서비스 분리를 위한 애플리케이션을 가상서버를 통해 분리 가능
- 빠른 서버 배포
 - 이미지 와 스냅샷을 통한 시스템 배포 및 복제 가능
- 재해복구
 - 스냅샷을 통한 가상서버 복원, VM 마이그레이션으로 데이터센터 이동 가능
- 동적 로드밸런싱
 - 작업 부하에 따라 자원 사용량 조절 가능
- 빠른 개발을 위한 테스트 환경제공
 - 가상서버를 통한 빠른 테스트 환경 제공으로 개발 시간 단축



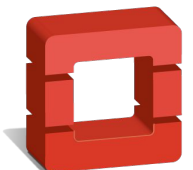
가상화의 장점

- 시스템 보안 과 신뢰성 향상
 - 가상서버와 물리적 하드웨어 사이 추상화 레이어를 통한 보안 강화
- OS 독립 , 하드웨어 벤더 락-인(lock-in) 축소
 - 하드웨어 추상화를 통해 가상 하드웨어는 벤더 락-인이 제거됨



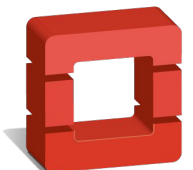
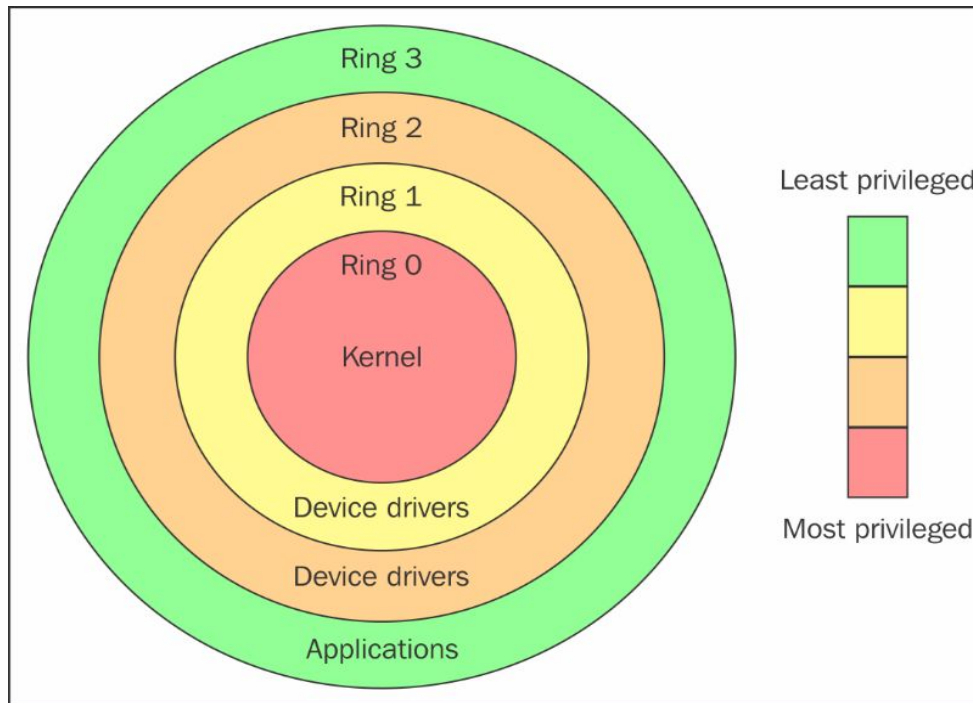
운영체제 가상화/파티셔닝

- 동일한 물리적 호스트가 서로다른 작업을 가능하게 함
 - 작업은 동일한 운영체제에서 독립적으로 작동
- 컨테이너 가상화
 - 물리적 서버에 다종의 격리된 운영체제 인스턴스(컨테이너)를 실행
- 컨테이너 가상화 종류
 - Solaris 컨테이너, FreeBSD jails, Parallels OpenVZ
- 단일 시스템에서 실행
 - 프로세스 격리 와 자원 관리는 커널이 담당
 - 컨테이너는 자신의 파일시스템, 프로세스, 메모리, 디바이스가 할당됨
- 멀티 OS 실행 제한
 - 윈도우, 리눅스, 유닉스등 다중 운영체제가 실행되는 가상화가 아님
 - 단일 OS 실행으로 성능과 효율성이 뛰어남



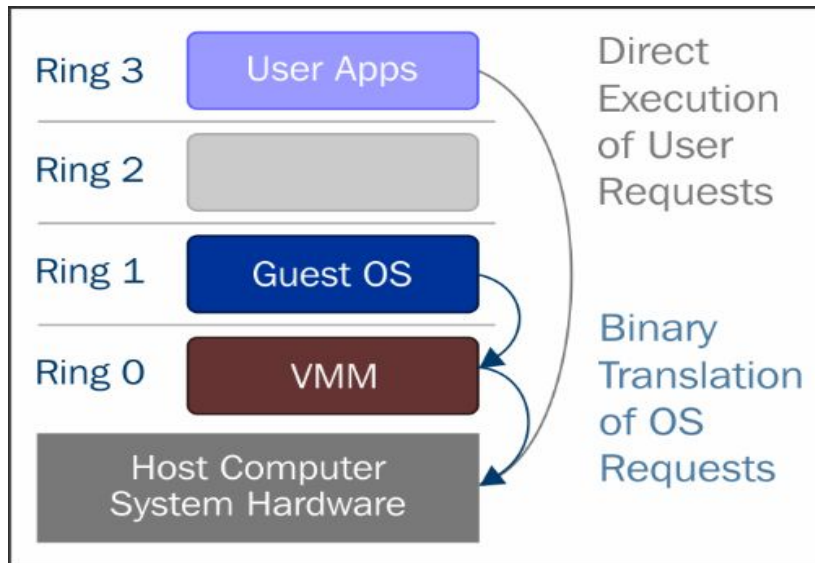
링 보호 (Protection Ring)

- 링 : 컴퓨터 시스템의 리소스 접근시 데이터 와 보안을 위한 장치
 - 링 0 : 커널모드 / 슈퍼 관리자 모드
 - 링 3 : 사용자 모드 / 애플리케이션 실행 모드



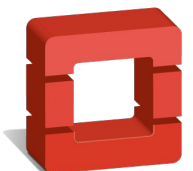
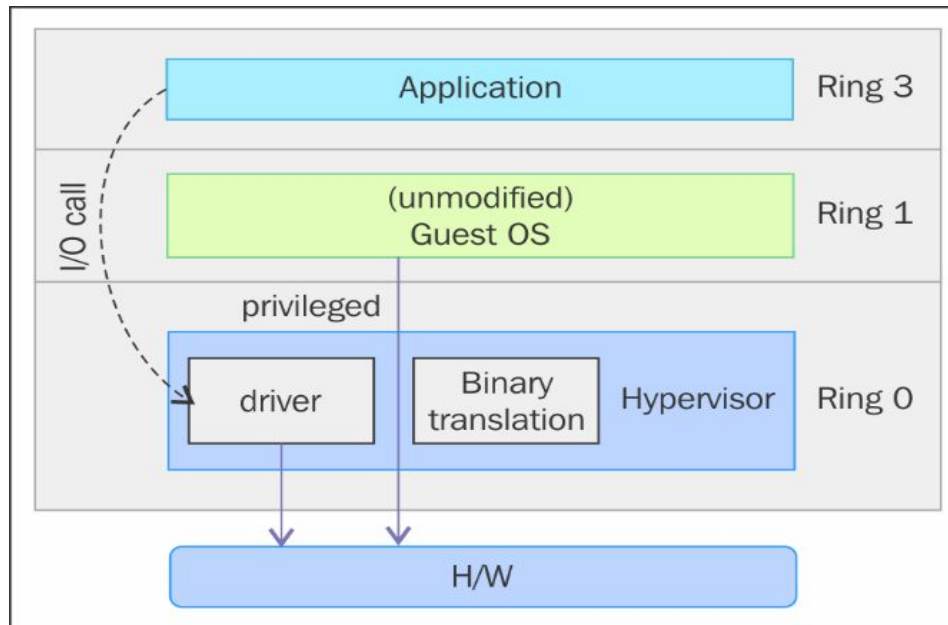
전 가상화 (Full Virtualization)

- Guest OS 는 링 1 에서 실행
 - OS 권한 제약 사항을 극복하기 위해 자원이 에뮬레이션 기능 사용
- 가상서버 모니터(VMM) 은 링 0 에서 실행
- 전 가상화는 x86 아키텍처에서 처음 실행됨
- 가상화 되지 않는 명령은 바이너리 변환 기술을 이용



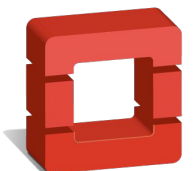
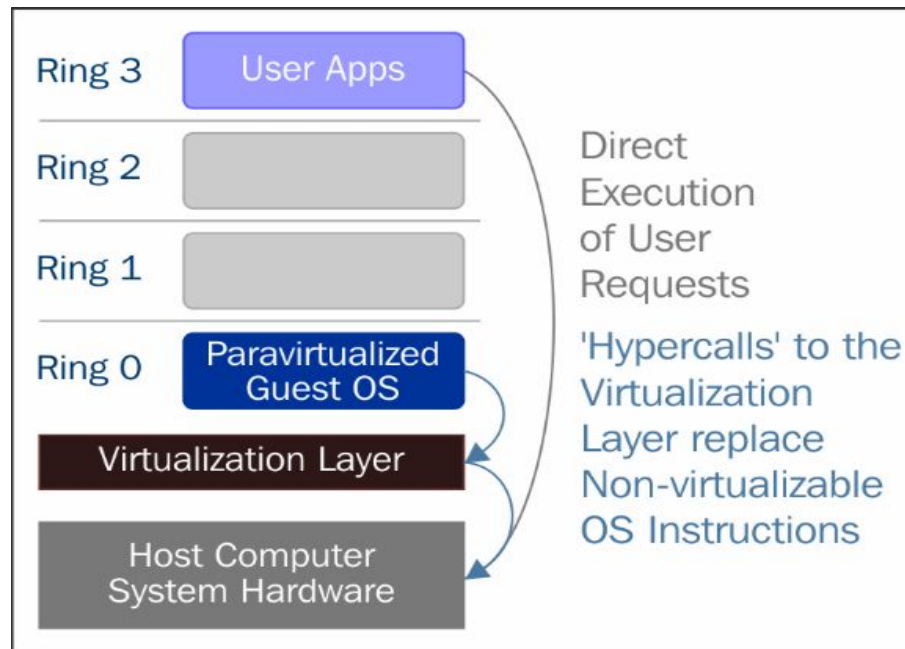
전 가상화 (Full Virtualization)

- 권한이 필요한 명령은 VMM 에 전달되어 바이너리 변환
- 바이너리 변환방식은 큰 성능 저하를 발생
- OS 의 커널 수정없이 사용 가능함
- VMM 은 CPU 를 관리하고 에뮬레이션 기능을 제공



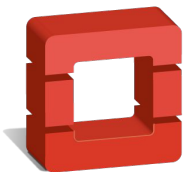
반 가상화 (Para Virtualization)

- Guest OS 링 0 에 접근 하기 위해 운영체제가 수정되어야 함
- Guest OS는 하이퍼바이저/VMM 사이에 하이퍼콜(Hypercall) 호출 실행
- 하이퍼바이저는 **API**를 제공
- 권한이 필요한 명령은 **API** 를 통해 실행 되어 링 0 에서 실행됨



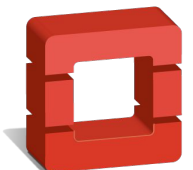
반 가상화 (Para Virtualization)

- Guest OS는 자신의 VM 이 가상화 되었음을 인지하고 있음
- 권한이 필요한 명령은 VMM 에 전달되기 위해 하이퍼콜 호출
- Guest OS 커널은 하이퍼콜을 통해 VMM 과 직접 통신 가능
- 바이너리 변환이 필요한 전가상화와 비교해 큰 성능 향상
- 반가상화 인지 가능한 특수한 Guest OS 커널이 필요



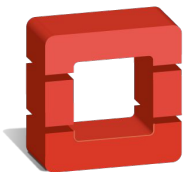
전가상화 vs 반가상화

구분	전가상화 (Full)				반가상화 (Para)			
	Host OS	Guest OS	Guest OS	Guest OS	Host OS	Guest OS	Guest OS	Guest OS
동작 방식	Hypervisor ↓ H/W ④ HW제어				Hypervisor ↓ H/W ③ HW제어			
		③ HW제어와 관련	① HW제어와 관련	② CPU 가상화 관련 등			① HW제어와 관련	② HW제어
설명	- CPU의 가상화 VT가 Hypervisor에게 HW제어의뢰				Guest OS가 Hypervisor에게 HW제어의뢰			
가상화 범위	- 하드웨어 전체 가상화				- 하드웨어 일부만 가상화			
성능	- HW제약 (CPU-VT 자원 부족)				- 고성능 유지 (System Call 없음)			
HW제어	- Guest OS가 직접 통제				- Hypervisor가 통제			
대상 OS	- Guest OS 수정없이 사용				- Guest OS 커널 일부 수정			
관련 제품	- VMware ESX Server				- Xen Express			



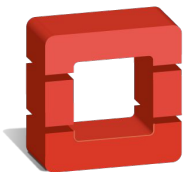
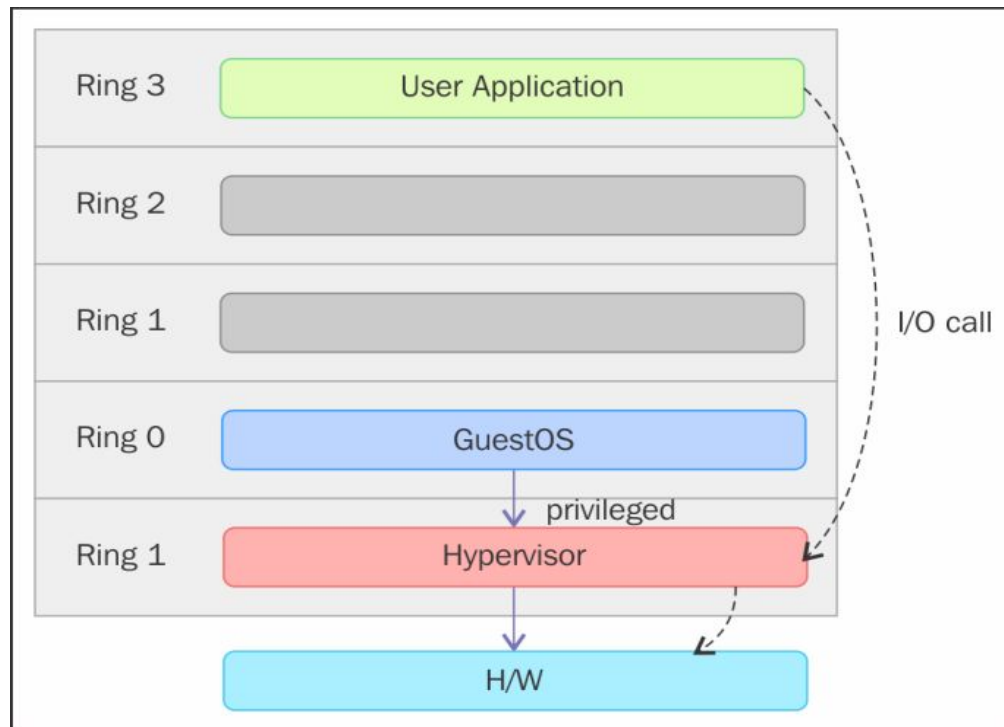
하드웨어 가상화 지원기능

- Intel 과 AMD 는 가상화가 x86 아키텍처의 중요 과제로 인식
- 프로세스 확장을 통한 독립적인 가상화 지원 기능 개발
 - Intel : Intel VT-x
 - AMD : AMD-V
 - Itanium : VT-i
- IA-32 명령어를 확장
 - Intel : VT(Virtualization Techonology)
 - AMD : SVM(Secure Virutal Machine)
- 하이퍼바이저와 VMM 이 Guest OS를 Ring 0 에서 실행 가능하도록 지원
- VM 을 링 0 에서 실행 가능 할 수 있도록, 하이퍼바이저 와 VMM 을 링 - 1 에서 실행함
- 기존 전가상화 기술보다 성능이 향상됨



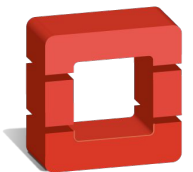
하드웨어 가상화 지원기능

- 하드웨어가 가상화를 인식
- 가상화 솔루션 개발시 복잡함을 줄여줌
- KVM 은 하드웨어 가상화 지원 기능을 사용



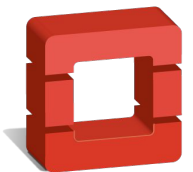
하이퍼바이저/가상서버 모니터(VMM)

- 하이퍼 바이저/가상서버 모니터(VMM)
 - VM , Guest OS 를 관리하는 소프트 웨어
- 가상화 작업 관리
 - 가상 하드웨어 제공
 - VM 생명주기, 실시간 리소스 할당, 가상머신 정책관리
- 가상서버 모니터 (VMM)
 - Guest OS 의 자원 할당 요청을 담당
 - 프로세스, 메모리등 시스템 하드웨어 환경 설정에 의해 자원 할당됨



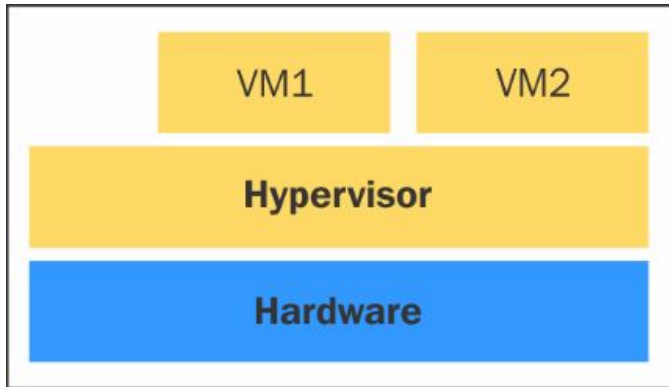
Type1 , Type2 하이퍼바이저

- 하이퍼바이저는 존재 위치에 따라 그 분류가 달라진다.
 - Type1 : 하드웨어 위에서 직접 실행
 - Type2 : 운영체제가 있고, 하이퍼바이저가 분리된 층에 실행
- Type1 하이퍼 바이저
 - 시스템 하드웨어와 직접 상호작용
 - 운영체제가 필요하지 않음
 - 베어메탈, 임베디드, 네이티브 하이퍼바이저라고 함
- Type2 하이퍼바이저
 - 운영체제 위에 존재
 - 다양한 변경이 가능하다
 - Hosted 하이퍼바이저



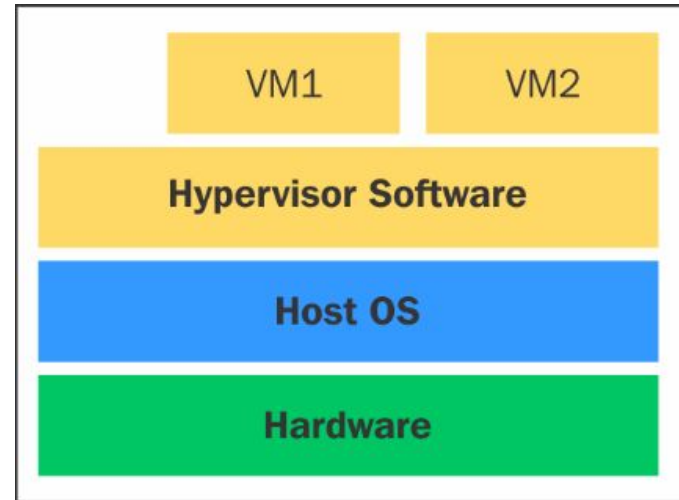
Type1 , Type2 하이퍼바이저

Type1



- 설치와 설정이 쉽다
- 사이즈가 작고 자원사용 최적화
- 부하가 작고 설치 애플리케이션이 작다
- 별도의 프로그램 및 드라이버 설치가 불가능

Type2

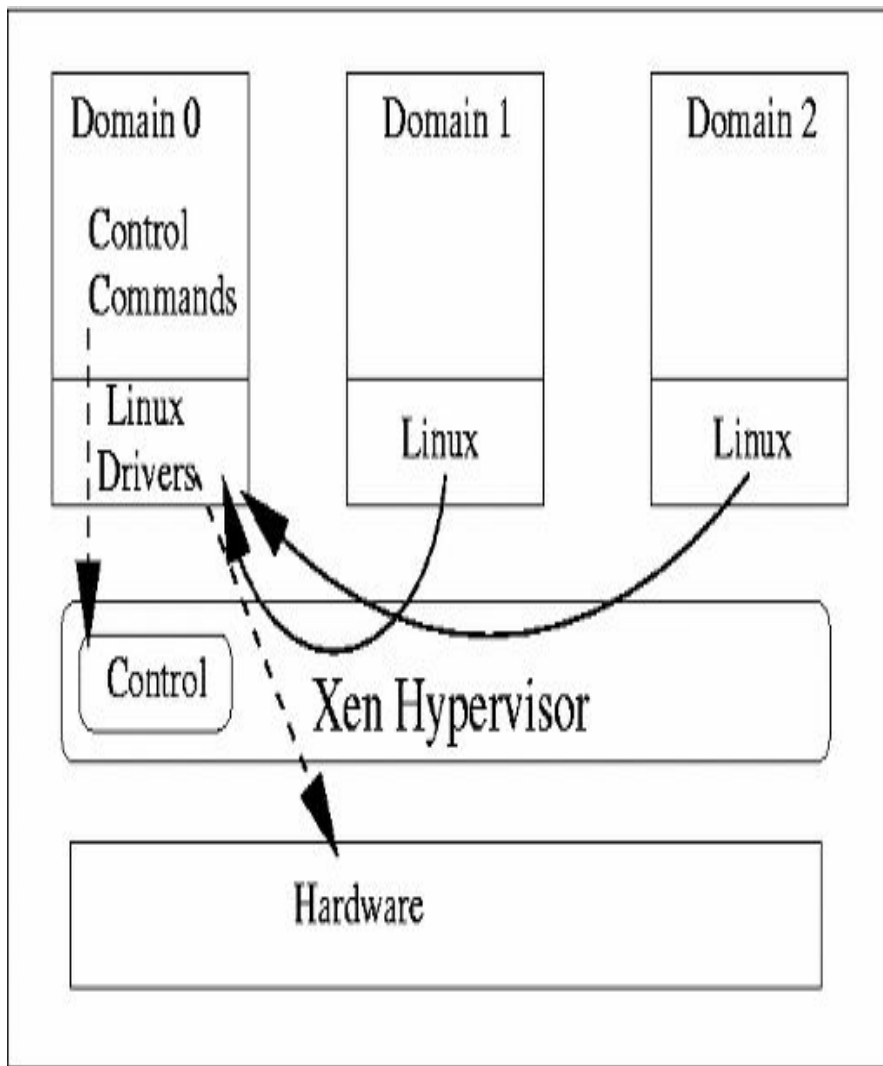


- 호스트 운영체제에 종속적이다.
- 광범위한 하드웨어 지원이 가능하다

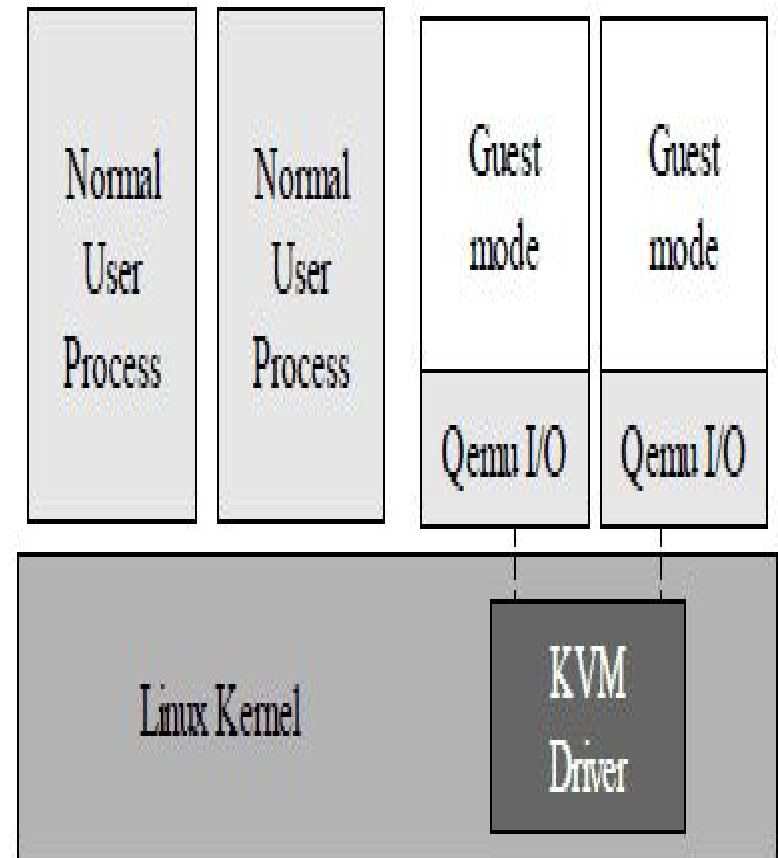
오픈소스 가상화 프로젝트

Project	Virtualization Type	Project URL
KVM (Kernel-based Virtual Machine)	Full virtualization	http://www.linux-kvm.org/
VirtualBox	Full virtualization	https://www.virtualbox.org/
Xen	Full and paravirtualization	http://www.xenproject.org/
Lguest	Paravirtualization	http://lguest.ozlabs.org/
UML (User Mode Linux)		http://user-mode-linux.sourceforge.net/
Linux-VServer		http://www.linux-vserver.org/ Welcome_to_Linux-VServer.org





XEN



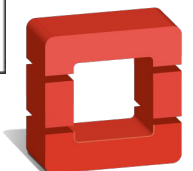
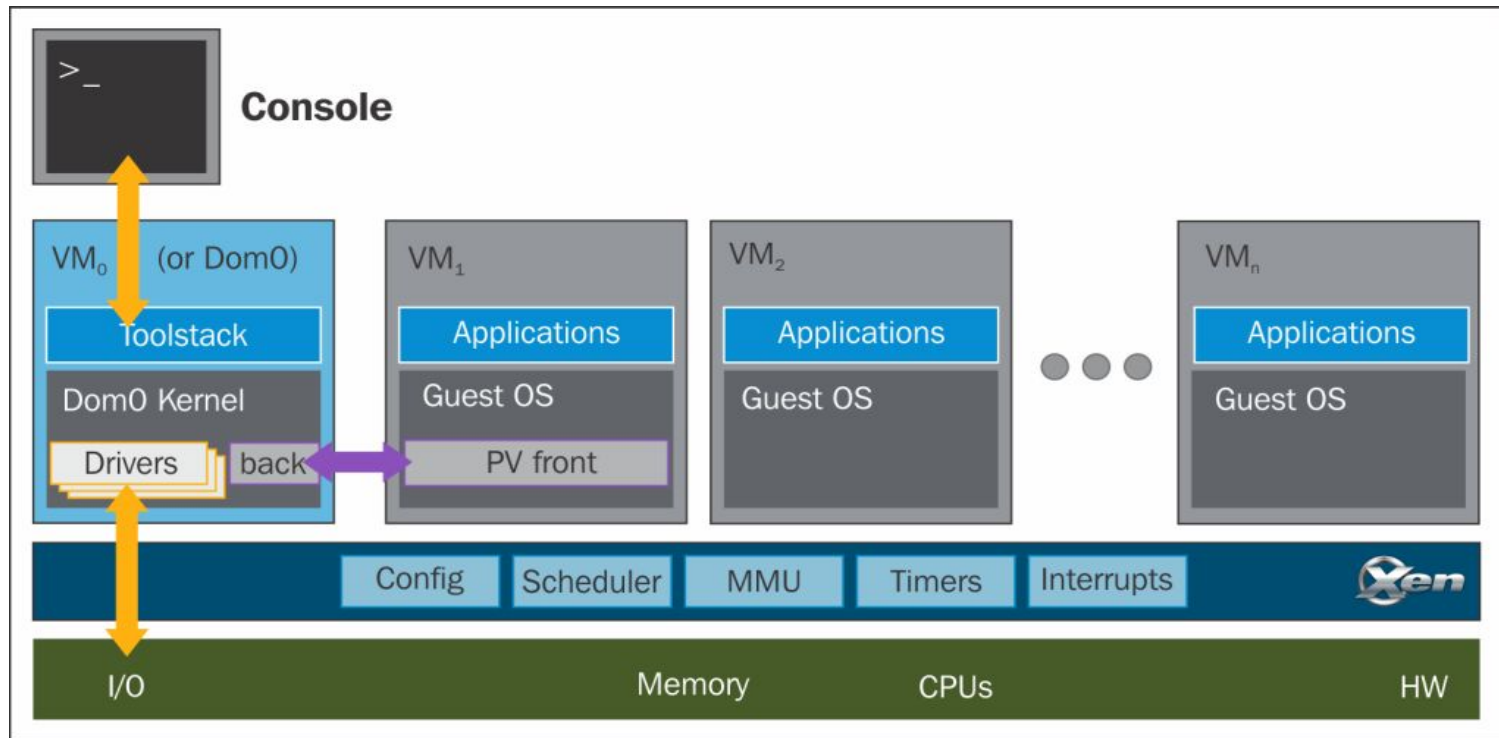
KVM



openstack™

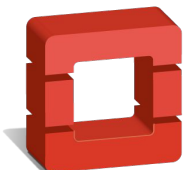
Xen

- 케임브리지 대학 연구 프로젝트로 시작, 2003년 공개
- 2013년 4월 Linux Foundation 공동 프로젝트 로 이전



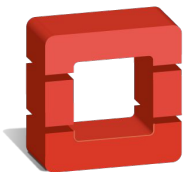
Xen

- 반가상화, 전가상화, 하드웨어지원 모드 사용 가능
- **Guest VM** 을 도메인 이라고 부른다.
- **Xen** 에는 두 종류의 도메인이 존재
 - **Dom 0** : 특권이 있는 도메인, 기능이 확장된 특수 **Guest VM**
 - **Dom U** : 권한이 없는 도메인, 일반 **Guest VM**
- **Dom 0**
 - VM을 만들고, 삭제하고, 관리 및 설정가능
 - 일반 **Guest** 시스템이 가상화 드라이버를 통해 하드웨어에 직접 접근 가능하도록 지원
 - **API** 인터페이스를 통한 시스템 관리기능 제공
 - 시스템에서 첫번째로 시작되는 도메인
 - **Xen** 프로젝트 하이퍼바이저를 위한 필수 도메인



KVM

- KVM (Kernel-based Virtual Machine)
- 하드웨어 가상화 지원기능 (VT-x, AMD-V) 를 활용한 최신 하이퍼바이저
- KVM은 KVM 커널모듈 설치로 하이퍼바이저로 변환가능
 - 리눅스 표준 커널에 KVM 커널 모듈을 추가
 - 표준 커널의 메모리 지원, 스케줄러 등을 사용가능한 장점이 있음
 - 리눅스 컴포넌트 최적화는 하이퍼바이저와 리눅스 Guest OS 둘다 이점을 갖음
- I/O 에뮬레이션을 위해 QEMU 를 사용
 - QEMU는 하드웨어 에뮬레이션을 지원하는 사용자 영역 프로그램
 - 프로세스, 디스크, 네트워크, VGA, PCI, USB, Serial/Parallel 포트 에뮬레이션



클라우드에서 리눅스 가상화가 제공하는것

리눅스는 클라우드 기반의 솔루션 개발을 위해 첫번째로 선택되어 왔다

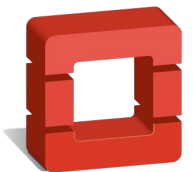
- 아마존 **EC2** 클라우드는 **Xen** 가상화를 사용
- 디지털오션 **KVM** 사용

리눅스 가상화를 사용한 오픈소스 **IaaS** 클라우드 소프트웨어

- 오픈스택
 - Openstack Foundation 에 의해 관리되는 오픈소스 **IaaS** 클라우드 솔루션
 - 몇개의 오픈소스 프로젝트 컴포넌트로 구성
 - KVM 을 기본 하이퍼바이저로 사용
- 클라우드스택
 - Apache Software Foundation 에 의해 관리되는 **IaaS** 클라우드 솔루션
 - 아마존 **EC2/S3 API** 와 호환성
 - Xen 을 기본 하이퍼바이저로 사용
- 유칼립투스
 - AWS 와 호환성을 가진 프라이빗 클라우드 소프트웨어
 - Xen, KVM 모두 지원



OpenStack VM



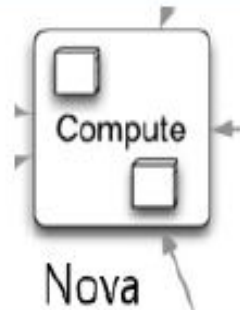
openstack™

IaaS

Virtualization
Server
Storage
Network



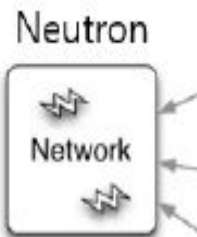
Server



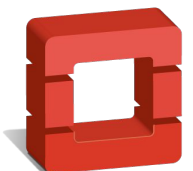
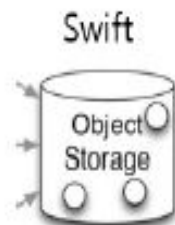
Virtualization



Network



Storage



openstack™



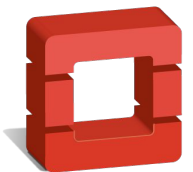
오픈스택의 기본 하이퍼바이저

KVM(Kernel Based Virtual Machine)

TYPE1(native or bare-metal)에 해당하는 하이퍼바이저

메모리 관리자는 파일 시스템 등과 같은 커널의 서브 모듈로 취급되며, KVM을 사용하기 위해서는 **CPU**에서 HVM(Hardware Virtual Machine) 기능을 제공해야 합니다.

HVM을 제공하는 **CPU**로는 x86 아키텍처의 Intel VT-x와 AMD의 AMD-v가 있습니다.



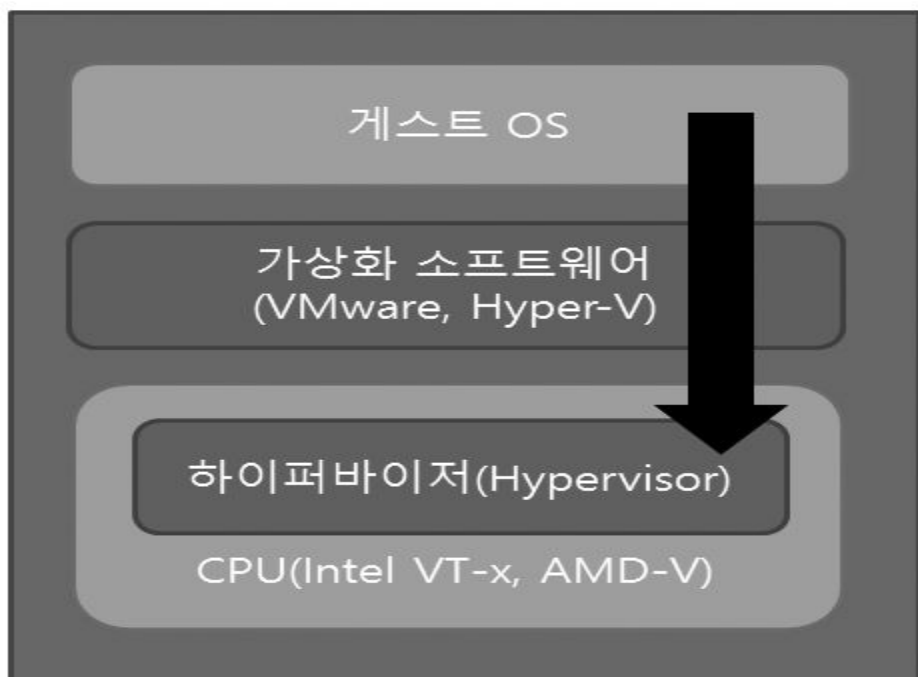
openstack™

가상머신과 Docker

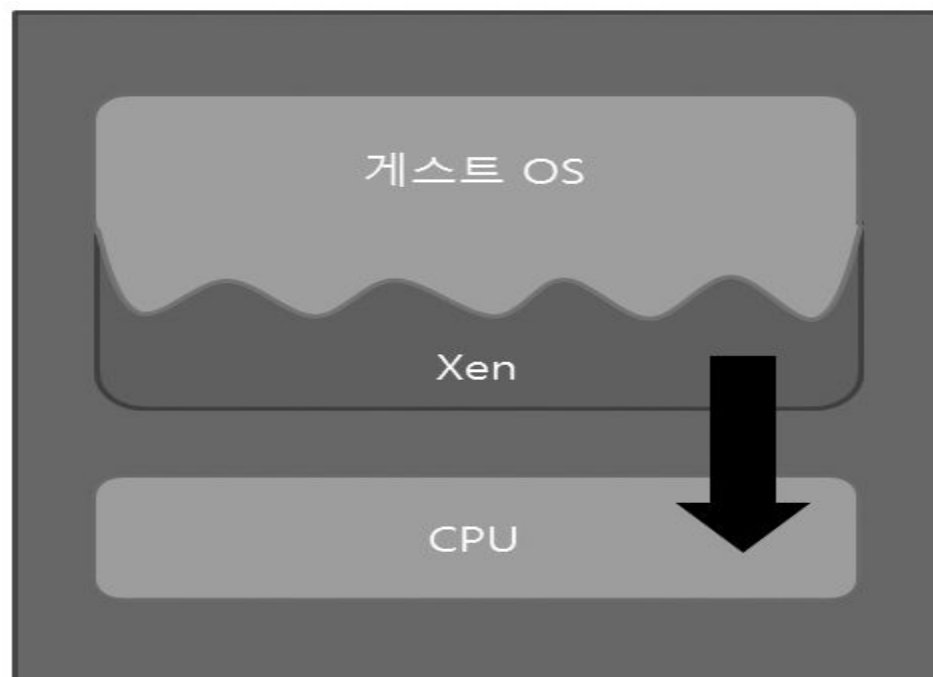


가상머신

전가상화(Full Virtualization)의 느린 속도를 개선하기 위해 반가상화(Paravirtualization) 방식이 개발되었고, 현재 널리 쓰이고 있습니다.



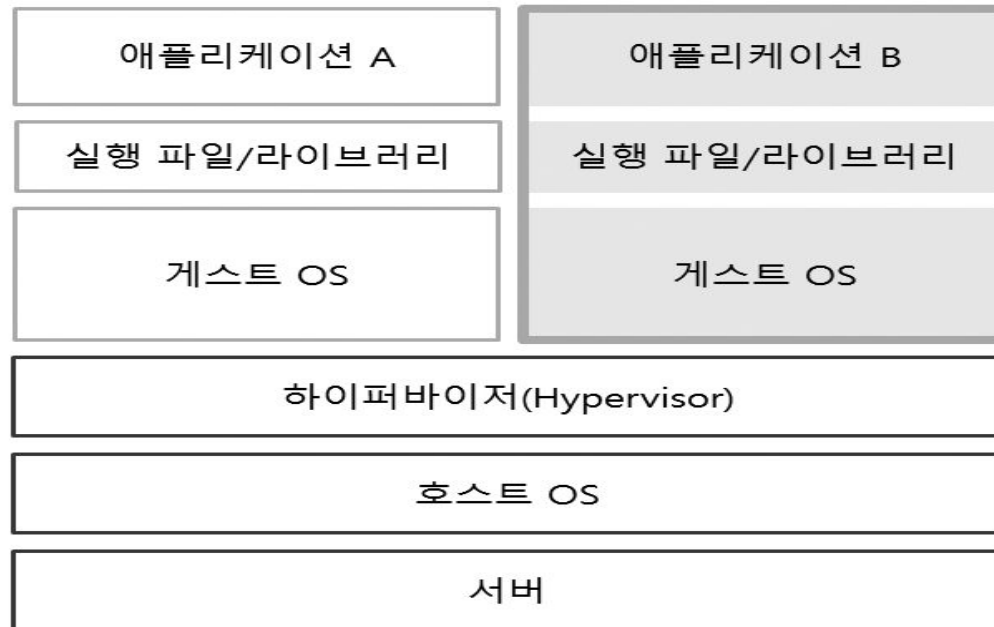
CPU의 가상화 지원을 이용한 전가상화



Xen 반가상화

가상머신의 한계

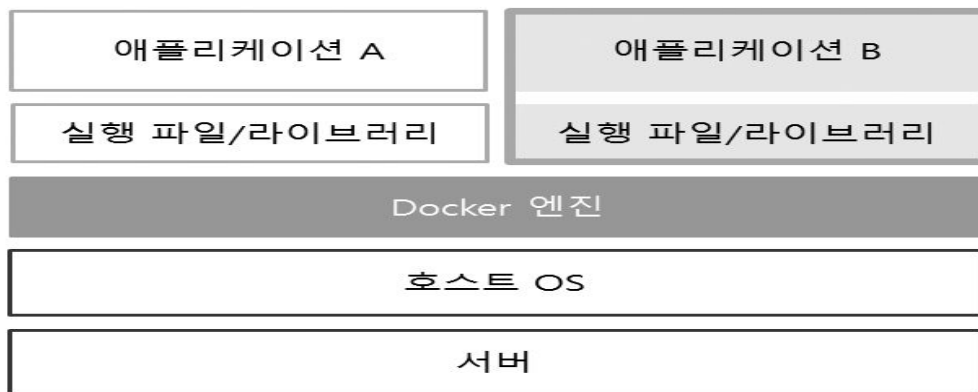
가상 머신 자체는 완전한 컴퓨터라 항상 게스트 OS를 설치해야 합니다.
그래서 **이미지 안에 OS가 포함되기 때문에 이미지 용량이 커집니다.**



네트워크와 인터넷 속도가 빨라졌다 하더라도 가상화 이미지를 주고받는 것은 꽤 부담스럽습니다. 특히 오픈 소스 가상화 소프트웨어는 OS를 가상화하는 것에만 초점이 맞춰져 있습니다. 그래서 이미지를 생성하고 실행하는 기능만 있을 뿐 **배포와 관리 기능이 부족**

Docker란?

Docker는 반가상화보다 좀더 경량화된 방식입니다. 그림 1-6와 같이 게스트 OS를 설치하지 않습니다. Docker 이미지에 서버 운영을 위한 프로그램과 라이브러리만 격리해서 설치할 수 있고, **OS 자원(시스템 콜)은 호스트와 공유**합니다. 이렇게 되면서 이미지 용량이 크게 줄어들었습니다.



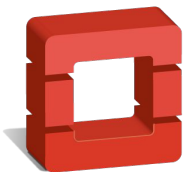
Docker는 하드웨어를 가상화하는 계층이 없기 때문에 메모리 접근, 파일시스템, 네트워크 속도가 가상 머신에 비해 월등히 빠릅니다.

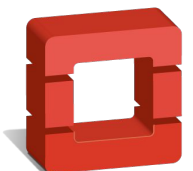
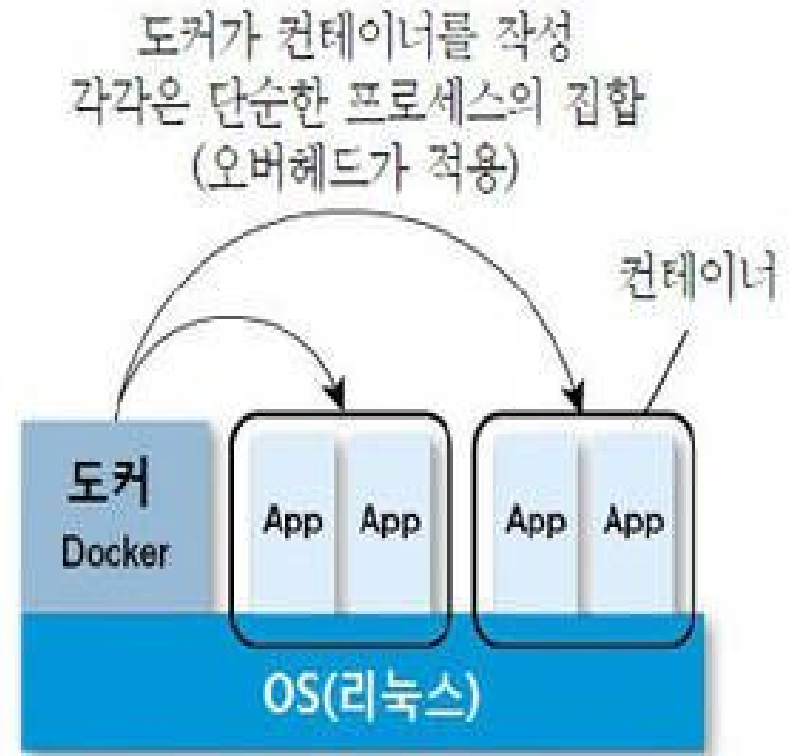
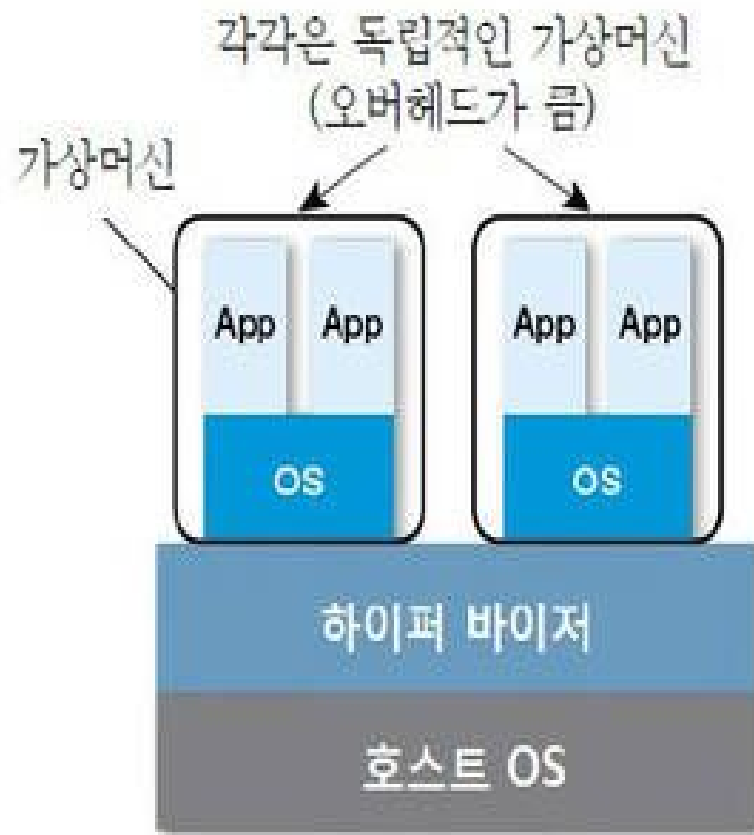
Docker의 장점

Docker는 가상 머신과는 달리 이미지 생성과 배포에 특화된 기능을 제공합니다.
Git에서 소스를 관리하는 것처럼 이미지 버전 관리 기능을 제공합니다.

또한, 중앙 관리를 위해 저장소에 이미지를 올리고, 받을 수 있습니다(**Push/Pull**).
그리고 **GitHub**처럼 **Docker** 이미지를 공유할 수 있는 **Docker Hub**도 제공합니다(**GitHub**처럼 유료 개인 저장소도 제공합니다).

다양한 **API**를 제공하기 때문에 원하는 만큼 자동화를 할 수 있어 개발과 서버 운영에 매우 유용합니다.





docker hub 사용

```
MINGW64 ~
$ docker search pyspark
time="2017-01-12T17:03:01.09:00" level=info msg="Unable to use system certificate pool: crypto/x509: system root pool is not available on Windows"
NAME                                DESCRIPTION                                STARS     OFFICIAL   AUTOMATED
thisgokeboysef/pyspark-docker      3                                             [OK]
prabeeshk/pyspark-notebook         3                                             [OK]
brillozon/pyspark-nb               Extends jupyter/pyspark-notebook with addi... 2         [OK]
opengovorg/pyspark                 OpenGov's base PySpark image with pre-inst... 2         [OK]
getwarped/pyspark-notebook         OpenShift compatible version of Jupyter pr... 1         [OK]
geraudster/pyspark-jupyter         Docker container with Jupyter and Spark      1         [OK]
fokkodriesprong/docker-pyspark     Docker container specially for running PyS... 1         [OK]
noieto/pyspark-jupyter            Jupyter Notebook Python + Spark for Toulou... 1         [OK]
korniichuk/pyspark                Apache PySpark                             0         [OK]
radanalyticsio/radanalytics-pyspark OpenShift S2I builder image for Python 2.7... 0         [OK]
produktion/jupyter-pyspark        Up to date jupyter-pyspark image            0         [OK]
vishnumohan/pyspark-notebook       pyspark-notebook                           0         [OK]
dpatriot/docker-pyspark-notebook   pyspark-notebook + sklearn libs             0         [OK]
rettydpi/pyspark-pytest            pyspark with pytest.                       0         [OK]
alexcoppe/pyspark                 Docker image to play around with PySpark.    0         [OK]
bluedne/pyspark                   Python 3 and spark                          0         [OK]
rettydpi/pyspark                  pyspark on docker.                         0         [OK]
audris/pyspark                    pyspark + rstudio + ldap                   0         [OK]
clipper/pyspark-container          Container for deploying PySpark models to ... 0         [OK]
fokkodriesprong/docker-pyspark-2   PySpark image build on top of Python 2      0         [OK]
publicisworldwide/data-science-pyspark Basic Data Science setup for Python workin... 0         [OK]
tmckay/pyspark-hdfs-notebook       This is a simple Python notebook image tha... 0         [OK]
smungee/pyspark-docker             0                                             [OK]
3blades/pyspark-notebook           PySpark Jupyter Notebook server             0         [OK]
josemyd/pyspark-notebook           PySpark notebook with Spark 2.0.0           0         [OK]

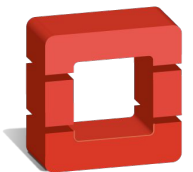
Home@LAPTOP-J94LGBTE MINGW64 ~
$
```

docker의 활용

syntaxnet-docker

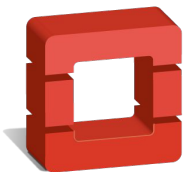
brat-docker

활용 사례

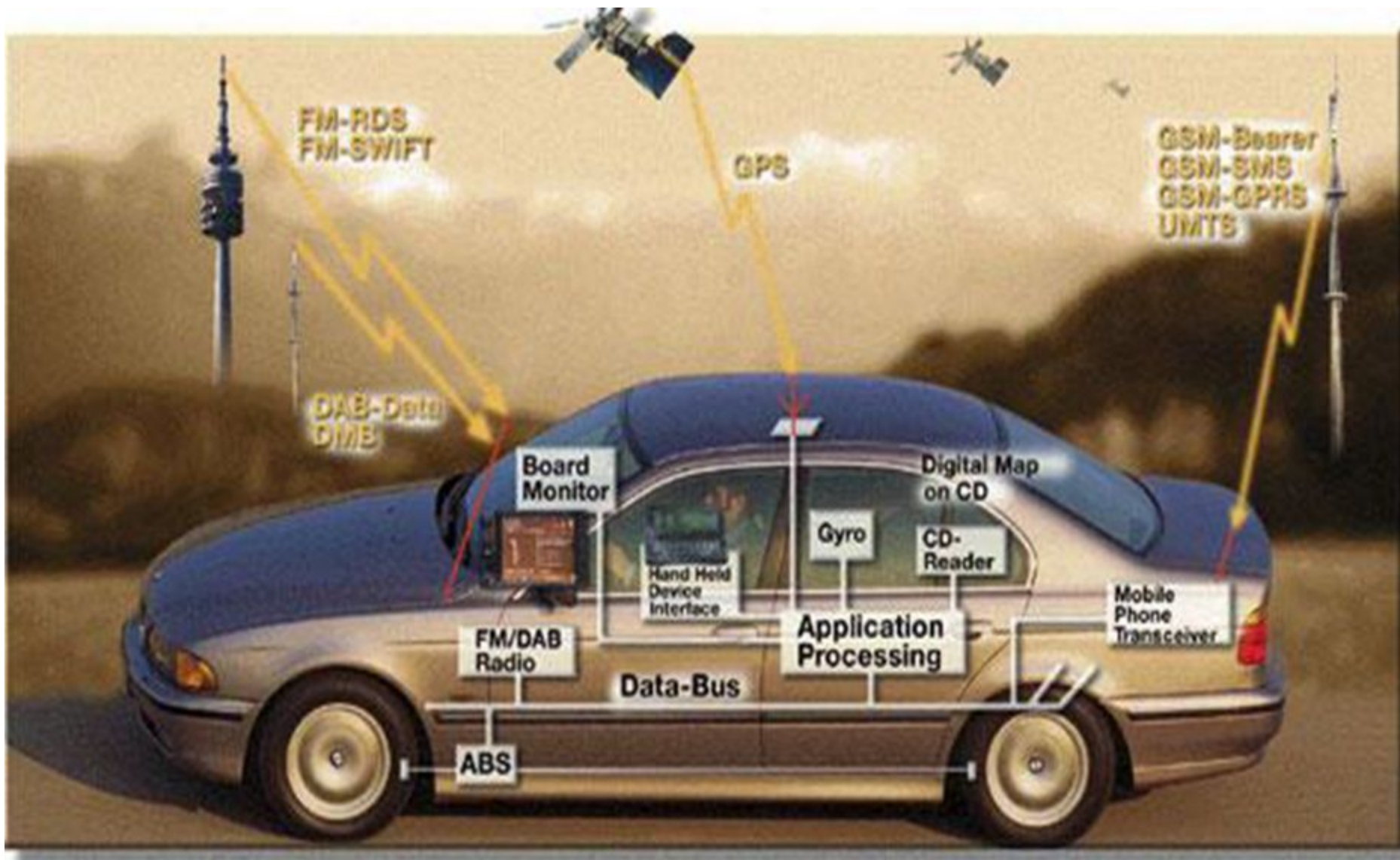


openstack™

Change your Perspective



openstack™



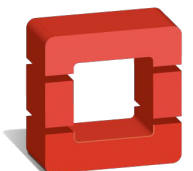
Automation Car

엔비디아의 차량용 슈퍼컴퓨터 ‘드라이브 PX2’는 자율주행차용으로 개발한 ‘파커(Parker)’ 프로세서와 ‘파스칼’ 아키텍처 기반 GPU가 각각 2개씩 들어간 단일 SoC(시스템 온 칩)를 탑재해 약 8TFlops의 연산능력을 갖췄다.

이를 통해 ▲차량 위치 파악 ▲안전한 운행궤도 측정 ▲360도 주변 상황 인식 등을 수행하면서 초당 약 24조 회의 딥러닝 작업 속도를 구현한다.

여기에 주변 환경을 보다 폭넓게 이해하기 위해 12개의 비디오카메라를 비롯해 레이더, 초음파 센서 등 다양한 경로에서 수집되는 정보들을 결합하는 ‘센서 융합’ 기술도 활용한다.

최신 테슬라 모델S(Tesla Model S)가 전시됐습니다. 이 모델은 DRIVE PX 2가 장착 된 최초의 시리즈 차량으로, DRIVE PX 2는 향후 소프트웨어 업데이트를 통해 완전한 자율성을 제공할 오토파일럿(AutoPilot) 시스템의 기반을 제공



openstack™

Automation Car



자동차 혁신을 지원하는 스타트업

출처 | 롤랜드버거



openstack™



삼성이 사들이는 하만그룹

설립 연도 1956년 직원 수 3만명

사업 분야 자동차 전자 장비, 소비자용 오디오,
기업용 음향·조명 기기 등

연매출 70억달러(약 8조2040억원).
전장 사업이 전체 65%

주요 보유 브랜드

JBL
JBL

AKG
by HARMAN
AKG
(프리미엄 오디오)

B&O
뱅앤올룹슨
(카오디오)

mark levinson
마크레빈슨

harman/kardon
하만카돈

주요 전장 사업 세계시장 점유율

- 카오디오(41%·1위)
- 정보·오락 장치(인포테인먼트·10%·2위)
- 자동차용 무선통신(텔레매틱스·10%·2위)

자료: 하만



openstack™

☀ LG이노텍 전장부품 수주현황

단위:조원 ■ 신규수주 ■ 수주잔고



☀ 2015년 2분기 중국 스마트폰 시장 점유율



☀ 2015년 2분기 세계 스마트폰 시장 점유율



☀ 자동차에 들어가는 다양한 전장부품



안전계

- 1 차선이탈 방지 시스템**
구성부품: 카메라, ECU
- 2 운전자 졸음 방지 시스템**
구성부품: IR카메라, IR LED, ECU
- 3 사각지대 감지 시스템**
구성부품: 카메라, 레이더, ECU
- 4 차량주변 모니터링 시스템**
구성부품: 카메라×4, ECU

편의계

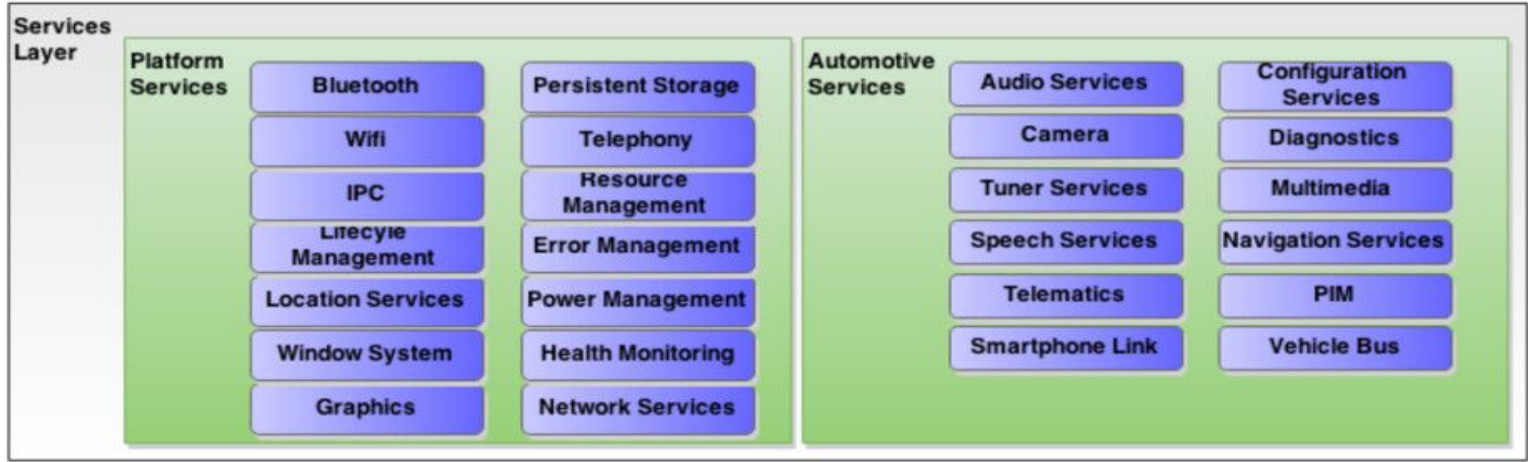
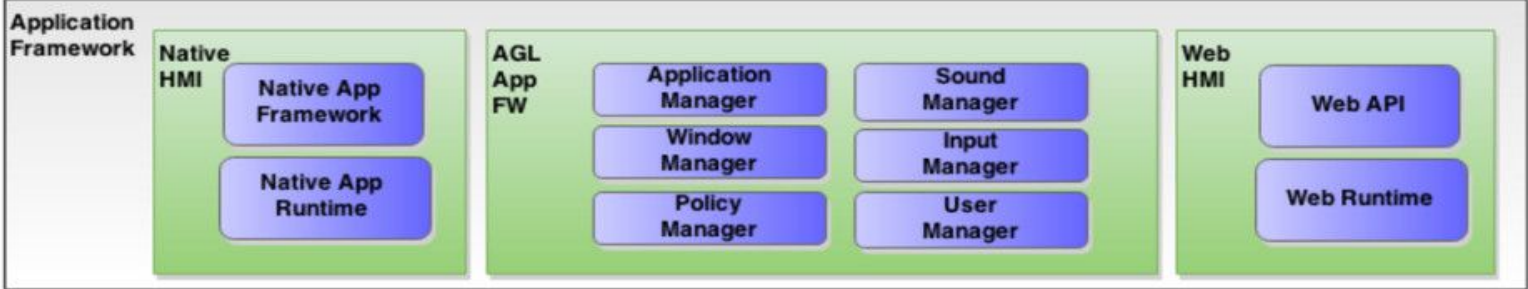
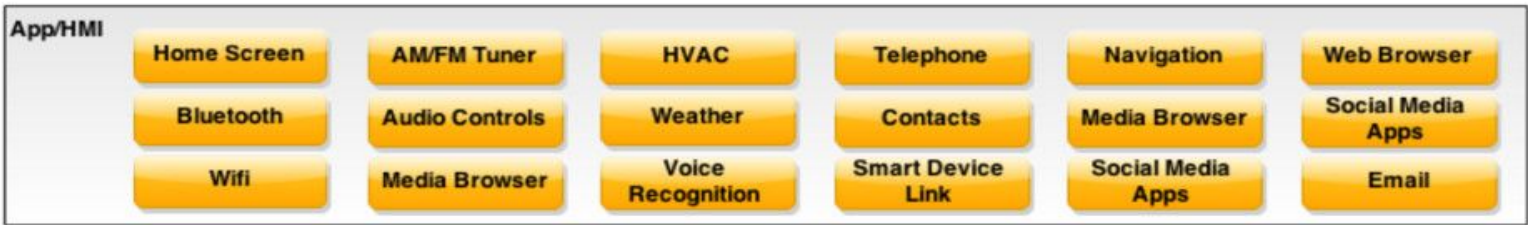
- 1 스마트기기 무선충전**
구성부품: 파워소자, 코일, IC
- 2 후방카메라**
구성부품: 렌즈, 이미지센서, 케이블

구동계

- 1 연비향상 보조장치(ISG)**
구성부품: 모터, 인버터, 컨버터
- 2 전기차용 충전기**
구성부품: 파워소자, 방열기구, 제어기
- 3 연비향상 보조장치(ISG)**
구성부품: 모터, 인버터, 컨버터

공용부품

- 수동부품**
구성부품: 세라믹, 전극, 유전체
- 자동차용 기관**
구성부품: 전열충, 구리
- 통신모듈**
구성부품: 반도체, SW, 패키지



Security

