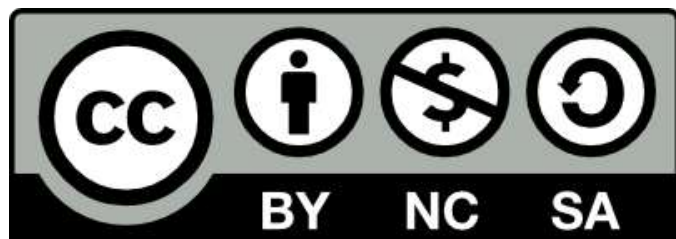


오픈소스의 이해

정명훈 (오픈소스컨설팅)



Google

facebook

NETFLIX



Go

NET



ook

오픈소스 기반 성공 사례 !!

오픈소스 기반 서비스 기업 사례



- 세계 최대의 오픈소스 기업
- 900여 개 프로젝트로 2,000만 라인 소스 공개
- Android
- Chromium
- GWT 등

facebook

- 200여 개 오픈소스 프로젝트 제공
- 1,000만 라인 소스 공개(2014/06 현재)
- React
- Hbase
- WebScaleSQL 등

Google, Facebook 만으로도
Linux 커널 두 개 분량

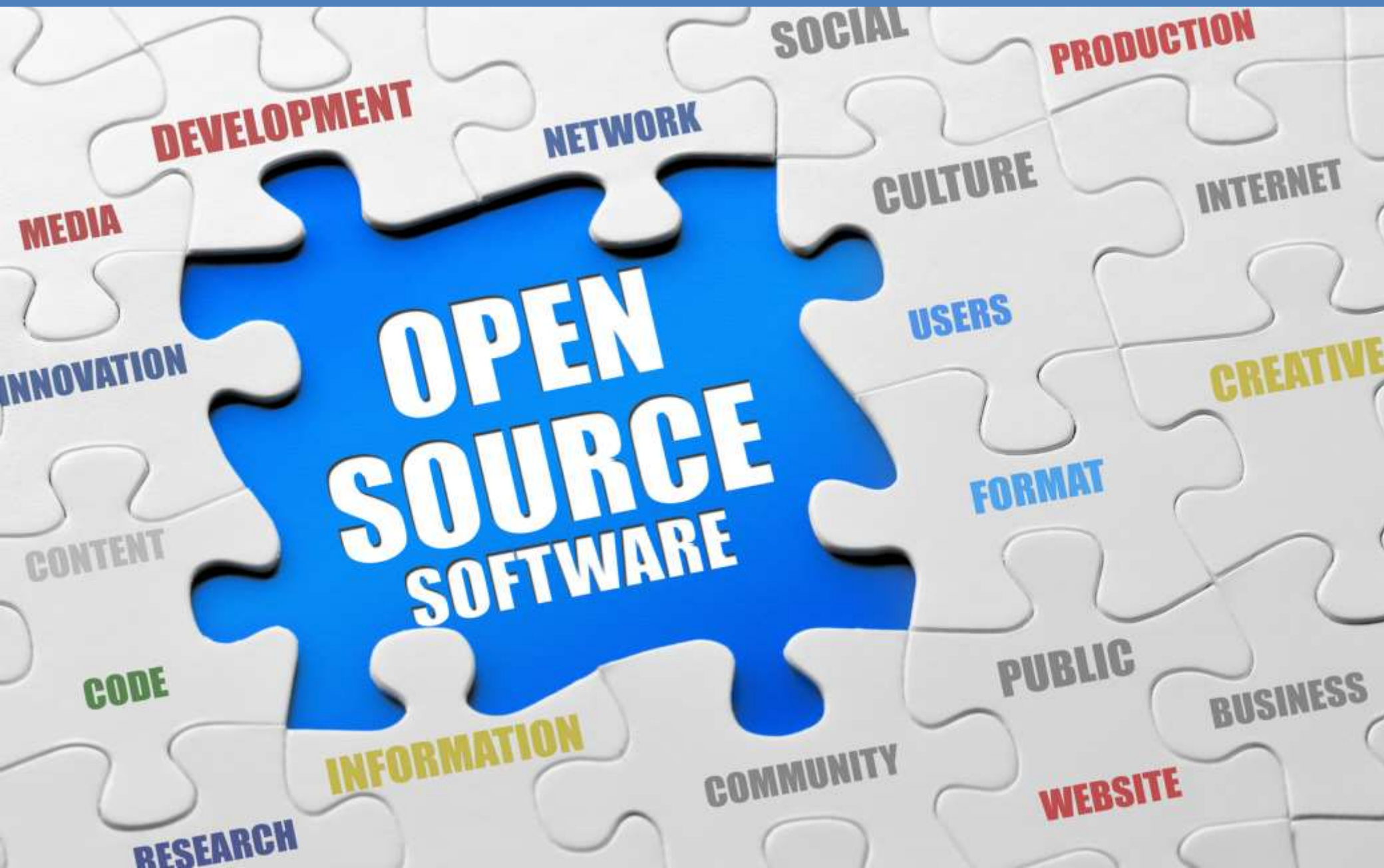
NETFLIX

- BigData, 개발/빌드, 데이터 저장소 등의 분야에서 수 십여 개 프로젝트 제공
- Lipstick
- Nebula
- RxJS 등



- "S2 Graph, 아파치 재단의 오픈소스 인큐베이터 프로젝트에 채택"
- MongoDB, OpenSSL, WebRTC 등 오픈소스를 활용하여 주요 서비스 개발

오픈소스란?



오픈소스란?

[위키피디아]

오픈 소스(open source)는 소프트웨어 혹은 하드웨어의 제작자의 권리를 지키면서 소스 코드를 누구나 열람할 수 있도록 한 소프트웨어 혹은 오픈 소스 라이선스에 준하는 모든 통칭을 일컫는다. 어원에 대해서는 History of the OSI 자료에 따르면, 1998년 2월 3일에 넷스케이프 브라우저의 원시 코드에 대해 어떠한 형태로 공개할까 하는 전략회의에서 붙여진 새로운 용어라고 설명되어 있다.

- 특정 라이선스에 따라 소프트웨어의 소스 코드가 공개되어 있음
- 일반적으로 오픈소스 소프트웨어에 대한 코드를 자유롭게 열람하거나 사용, 복사, 수정, 개작 및 재 배포의 권한("자유", Freedom)을 부여 받음
- Linux 운영체제가 대표적인 오픈소스 사례로, 누구나 무료로 이용할 수 있으며 공개된 코드를 기반으로 프로그램을 마음대로 변형 할 수 있음
- 오픈소스는 소프트웨어뿐 아니라 하드웨어에도 적용 됨
- 사용 시, 원 제작자의 권리를 지키는 것도 중요 함

OSI(Open Source Initiative)에서 말하는 오픈소스 조건

1. 자유로운 재배포

- 라이선스는 여러 다른 소스들의 프로그램을 포함하고 있는 집합적 소프트웨어 배포판의 컴포넌트로서 소프트웨어의 판매나 기부에 대해서 그 어떤 누구도 배제되어서는 안 된다. 라이선스는 로열티 또는 판매 비용이 없다.

2. 소스 코드

- 프로그램에는 소스 코드가 포함되어 있어야 하며 소스 코드 형태와 컴파일이 완료된 형태로 배포될 수 있어야 한다. 특정 형태의 제품이 소스 코드로 배포되지 않으면 소스 코드를 구매해야 하며 인터넷을 통해 무료로 다운로드 해야 한다. 소스 코드는 프로그래머가 프로그램을 변경하기에 좋은 형식이다. 알아보기 힘든 소스 코드는 사용될 수 없다. 프리프로세서(preprocessor)의 아웃풋 또는 트랜슬레이터 같은 중간 형태도 사용될 수 없다.

3. 파생 작업(Derived work)

- 라이선스는 변경과 파생 작업이 가능해야 한다. 원래의 소프트웨어 라이선스와 같은 조건 하에 배포되어야 한다.

4. 소스 코드의 무결성

- 구현 시 프로그램을 변경할 목적으로 소스 코드와 함께 "패치 파일"의 배포를 허용하는 경우에만, 라이선스는 소스 코드가 변경된 형태로 배포되는 것을 제한한다. 라이선스는 변경된 소스 코드에서 구현된 소프트웨어의 배포를 허용해야 한다. 라이선스는 파생 작업(derived work)을 통해 다른 이름 또는 다른 버전을 만들어야 한다.

5. 개인 또는 그룹의 평등

- 라이선스는 어떤 개인이나 그룹에 차별을 두어서는 안 된다.

OSI(Open Source Initiative)에서 말하는 오픈소스 조건

6. 분야에 대한 평등

- 라이선스는 특정 분야에서 프로그램을 사용하는 것에 대해 제한을 두어서는 안된다. 예를 들어, 프로그램은 비즈니스 또는 유전공학 연구 분야에서 사용될 수 있다.

7. 라이선스 배포

- 프로그램에 대한 권한은 프로그램이 재배포 된 모든 곳에 적용되어야 한다. 추가 라이선스를 발행할 필요가 없다.

8. 제품 스펙에 따른 라이선스

- 프로그램에 대한 권한은 특정 소프트웨어 배포판의 프로그램의 일부에 해당하지 않는다. 프로그램이 그 배포판에서 추출되었고, 그 프로그램의 라이선스의 조건 하에 사용 및 배포된다면 재배포된 프로그램을 사용하는 모든 당사자들은 원래의 소프트웨어 배포판에서 허용된 것과 같은 권한을 갖게 된다.

9. 라이선스는 다른 소프트웨어를 제한하지 않는다

- 라이선스는 라이선스를 받은 소프트웨어와 함께 배포된 다른 소프트웨어에 제약 사항을 두어서는 안된다. 예를 들어, 같은 미디어에 배포된 모든 다른 프로그램들이 오픈 소스 소프트웨어가 될 필요는 없다.

10. 라이선스는 기술 중립적이어야 한다

- 라이선스는 기술이나 인터페이스 스타일을 한정해서는 안된다.



IT 역사에서 오픈소스

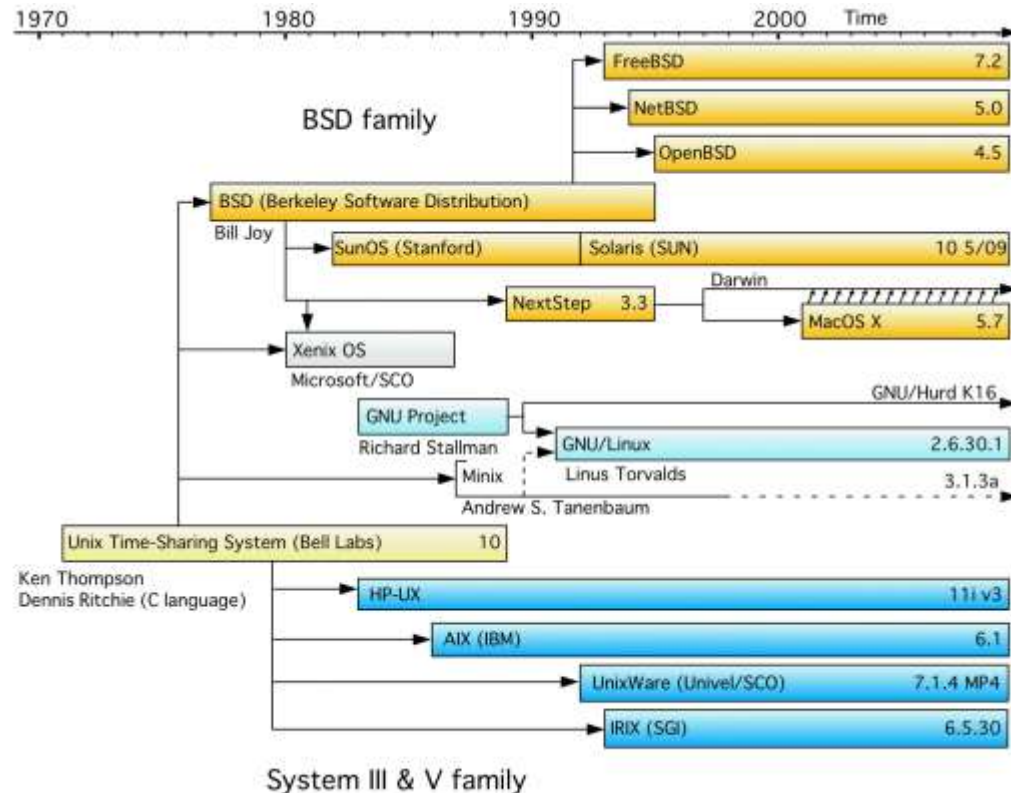
- 1960년대
 - 하드웨어 판매 중심
 - 소프트웨어 공유 문화



Unix 개발 및 상용화

● 1970년대

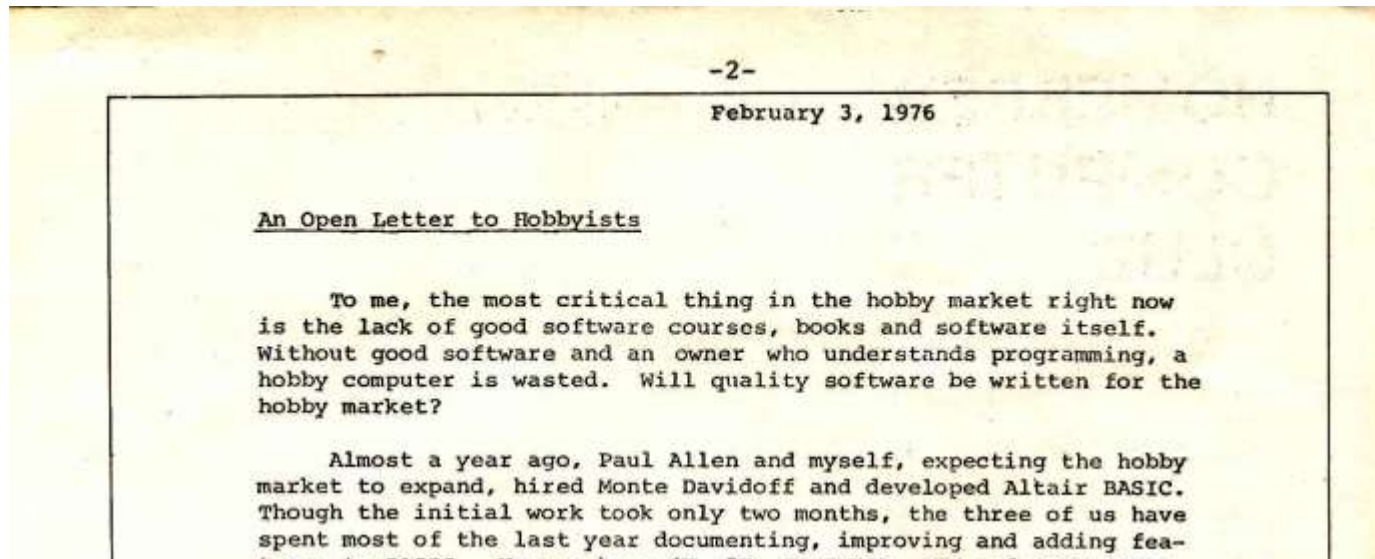
- 벨 연구소에 의해 Unix 개발
- AT&T를 비롯한 여러 회사, 버클리 등 비영리 단체에서 다양한 버전 개발
- 1980년대 HP/UX, AIX, Solaris 등의 상용 버전



상용 소프트웨어의 등장

● 1976년

- 빌게이츠가 컴퓨터 애호가들에게 보내는 공개 편지 "Open Letter to Hobbyists"
- 1976년 친구 폴 앨런과 MITS Altair 8800을 위한 베이식 인터프리터 제조 후 MITS를 통해 판매
- 소프트웨어 저작권에 대한 인지, 시작된 상업용 소프트웨어 시대
- 모든 코드를 비공개 형태로 만들어 버리게 되는 계기



자유소프트웨어 운동

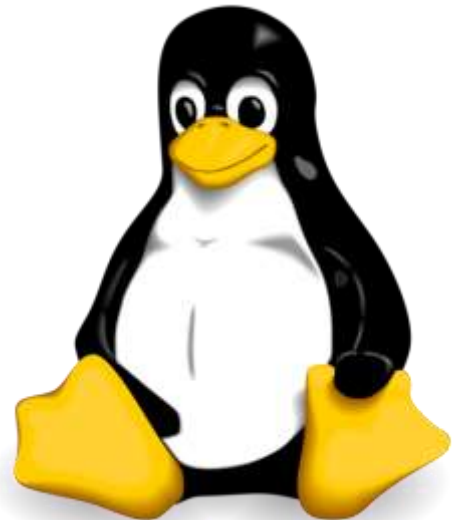
● 1984 ~ 1986년

- 리처드 스톨만
- 자유 소프트웨어 운동 주도
- 상용 소프트웨어에 대항한 자유로운 대안을 만들기 위한 의지
- GNU is Not a Unix(GNU) 프로젝트 시작
- Text editor(Emacs), Compiler(gcc), Debugger(gdb), make 개발
- Free Software 운동 시작 후 자유소프트웨어 재단(FSF) 설립
- 1989년 최초의 오픈소스 라이선스 GNU General Public License 배포



Linux의 탄생

- 1991년
 - 리눅스의 탄생
 - 핀란드 헬싱키 대학의 대학원생인, 리누스 토발즈가 취미로 개발한 커널
 - 핀란드 대학 3학년, MINIX를 응용하여 리눅스 커널 개발 후 GNU GPL로 배포
- 1992년
 - Linux Desktop GUI 프로젝트, Xfree86 시작
- 1993년
 - GNU/Linux 구현체 Debian, Slackware 시작



오픈소스라는 이름

● 1998년

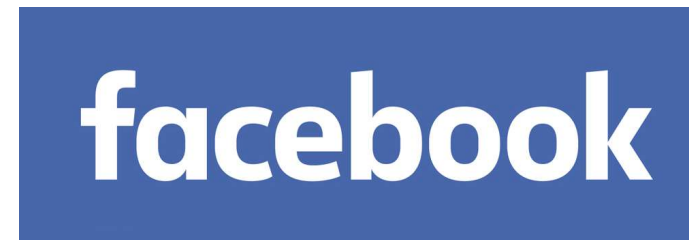
- 넷스케이프 브라우저 코드에 대한 공개 형태를 결정하는 전략회의에서 붙여진 용어
- 넷스케이프 소스 코드 공개에 자극 받아, 오픈소스를 장려하기 위한 단체인 OSI(Open Source Initiative) 설립



오픈소스의 발전

- 2000년대 이후

- 많은 기업들이 오픈소스를 사용
- 상용 소프트웨어 벤더에서도 오픈소스를 가져다가 사용하기 시작
- 특히, 웹 기반 서비스(예: SNS) 업체에서 오픈소스 활용 폭발적 증가
- 구글, 아마존, 넷플릭스, 인텔, 오라클, IBM 등





오픈소스의 현재

오픈소스의 확대 – 이제는 거의 모든 분야

운영체제



웹서버/미들웨어



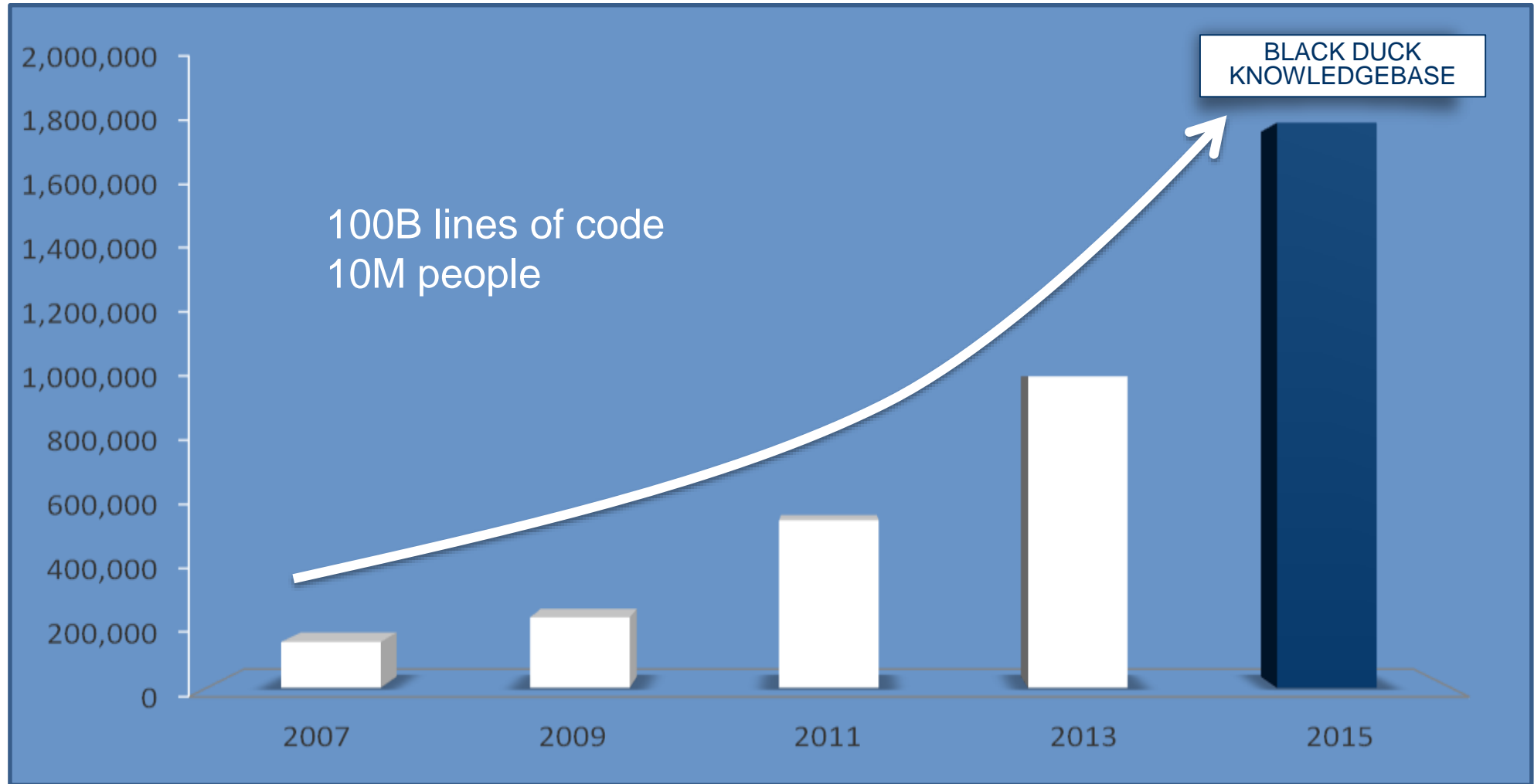
데이터베이스



기타



오픈소스의 확대 – 프로젝트와 참여의 비약적 증가



Source: 2014 Future of Open Source Survey Results

오픈소스는 IT 업계의 대세...



78%의 회사가 오픈소스를 사용 중
64%의 회사가 오픈소스 프로젝트 참여 중

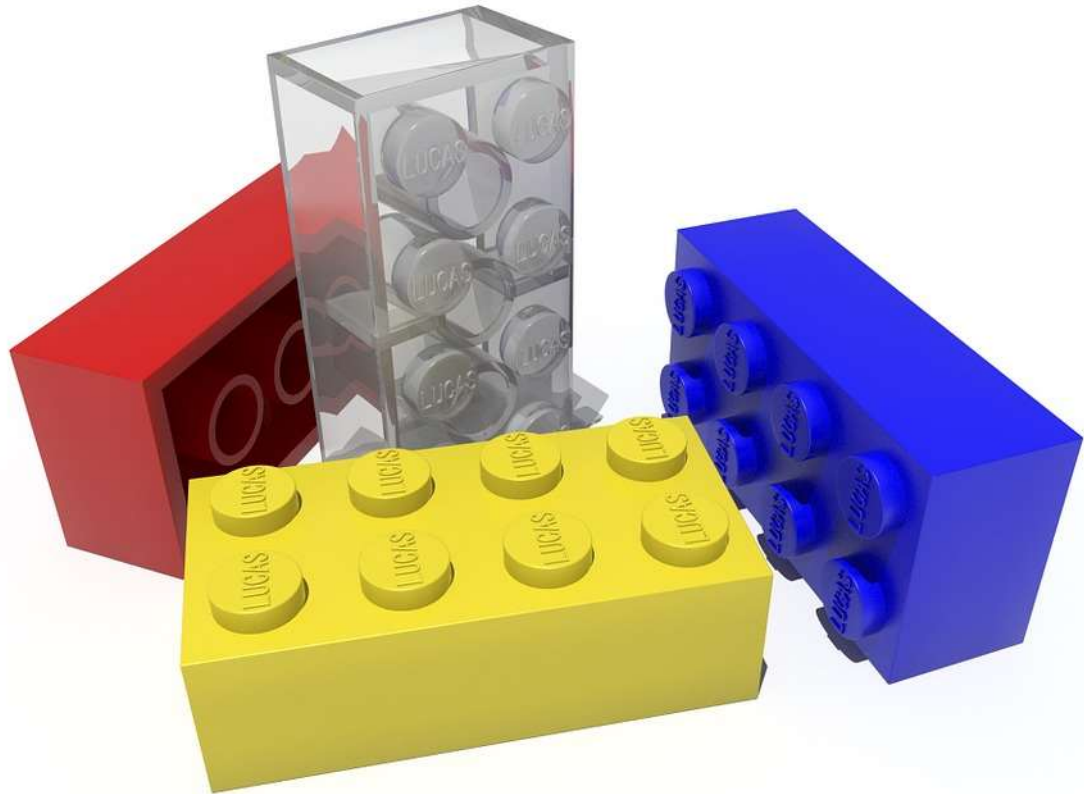
Source: 2015 Future of Open Source Survey Results



오픈소스 블록으로 원하는 것을 쉽게 구현...

50% 이상의 엔지니어가 오픈소스 기반으로 업무
66% 이상이 솔루션 도입 시 오픈소스 우선 고려

Source: 2015 Future of Open Source Survey Results



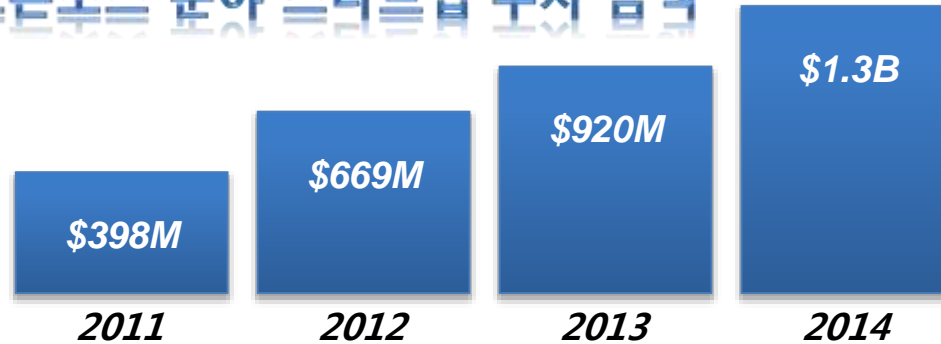
오픈소스는 기업의 비즈니스를 돕고 있는 중...



65%의 기업이 오픈소스를 통해 경쟁 우위 확보
75%의 기업이 오픈소스를 통해 매출 증대

증가하는 오픈소스 스타트업에 대한 투자

오픈소스 분야 스타트업 투자 금액



Source: 2015 Future of Open Source Survey Results



START
UP





오픈소스 현황

Apache Software Foundation (오픈소스 재단)

APACHE PROJECT LIST

BY CATEGORY

Overview
All Projects
Attic
Big Data
Build Management
Cloud
Content
Databases
FTP
Graphics
HTTP
HTTP-module
Incubating
JavaEE
Labs
Libraries
Mail
Mobile
Network-client
Network-server
OSGi
RegExp
Retired
Testing
Virtual-machine
Web-framework
XML
FAQ

BY NAME

HTTP Server
A
Abdera
Accumulo
ACE
ActiveMQ
Airavata
Allura
Ambari
Ant
Any23
APR
Archiva
Aries
Aurora
Avro
Axis

Commons
Continuum
Cordova
CouchDB
Creadur
Crunch
cTAKES
Curator
CXF
D
DB
DeltaSpike
DeviceMap
Directory
Drill
Flex
Flink
Flume
Forrest
G
Geronimo
Giraph
Gora
Groovy
Gump

H
Hadoop
Hama
HBase
Helix
Hive
HttpComponents
I
Isis
Ignite
J
Jackrabbit
James
K
Kafka
Karaf
Knox
Kylín
L
Lens
Libcloud
Logging
Lucene
Lucene.Net
Lucy

M
Mahout
ManifoldCF
Marmotta
Maven
Mesos
MetaModel
MINA
MRUnit
MyFaces
N
Nifi
Oni
Onami
OODT
Oozie
Open Climate
Workbench
OpenJPA
OpenMeetings
OpenNLP
OpenOffice
OpenWebBeans
ORC
P
Parquet
PDFBox
Perl
Phoenix
Pig

Pivot
POI
Portals
Q
Qpid
R
Rave
REEF
River
Shiro
SIS
Sling
SpamAssassin
Spark
Sqoop
Stanbol
STeVe
Storm
Stratos
Struts
Subversion
Synapse
Syncope
T
Tajo
Tapestry
Tcl

Tez
Thrift
Tika
Tiles
Tomcat
TomEE
Traffic Server
Turbine
U
V
W
Velocity
VXQuery
Web Services
Whimsy
Wicket
Wink
Wookie
X
Xalan
Xerces
XMLBeans
XML Graphics
Y
Yetus
Z
Zest
ZooKeeper

2016년 현재 약 350개 이상의 오픈소스 프로젝트 관리

Apache Software Foundation (오픈소스 재단)

- Apache Software Foundation(ASF)
 - ASF 이전에는 "The Apache Group"
 - 정형화되고 법률적인 지원을 받기 위해 설립(1999)
 - 초기에는 21명의 멤버, httpd 웹서버 프로젝트 하나로 시작
 - Apache Jakarta 프로젝트를 통해 Tomcat, Struts, Ant, Maven 등 각종 Java 기반 오픈소스 시작
 - 운영에 필요한 서버와 서비스는 기부 받음
- 왜 재단이 만들어질까?
 - 프로젝트를 할 수 있는 기본적 인프라 제공
 - 컨퍼런스, 이벤트, 기부금 등 세금 관련된 일들 처리
 - 라이선스, 특허, 표준 약관, 상표 관리 등 법률적 문제 대응
 - 기술 중심 워크그룹 운영



주요 오픈소스 재단

- Apache Software Foundation

- 가장 대표적인 오픈소스 재단. 350개 이상의 프로젝트 관리



- Software Freedom Conservancy

- BusyBox, Git, Samba, Wine 등 33개 프로젝트 관리
- GPL 의무화를 위한 GPL Compliance Project를 추진



- Linux Foundation

- Linux Kernel 관리. 최근 SDx, IoT, Embed, Cloud 등의 다양한 이머징 기술 협력



- Eclipse Foundation

- 이클립스 개발툴을 비롯한 배포, 관리 오픈소스 관리
- BI(Business Inteligence), IoT 등에 관한 200여 개 프로젝트 지원



주요 오픈소스 재단

- Cloud Foundry Foundation

- 클라우드 구축에 필요한 인프라 소프트웨어들을 관리
- EMC, HP, IBM, 인텔, SAP 등의 벤더가 공동 참여



- Free Software Foundation

- 리처드 스톨만이 창립했으며, 소프트웨어의 자유로운 사용에 초점을 맞추고 있음
- 무료 운영 시스템 구현에 목표를 둔 GNU 프로젝트만 관리



- Open Source Initiative

- 역시 특정 프로젝트가 아닌 '오픈소스 소프트웨어' 운동을 지원



- Openstack Foundation

- 오픈스택 클라우드(Public Cloud) 운영 시스템 프로젝트



산업별 오픈소스 커뮤니티 – 오픈소스 기반의 2차 혁명



- XENIVI
 - 차량용 인포테인먼트 오픈소스 플랫폼 채택을 목표로한 단체
 - 160개 회원사(현대, BMW, 혼다 등 완성차 업체 포함) 가입
- OSHERA
 - EHR(Electronic Health Record) 등 의료용 소프트웨어를 오픈소스 기반으로 제공하기 위한 단체
- OpenMAMA
 - 미들웨어나 벤더 중립적인 메시징 API를 구현하기 위한 단체
- Allseen Alliance
 - 퀄컴과 Linux Foundation이 사물인터넷용 오픈 플랫폼과 표준을 제정하기 위한 단체
- Polarsys
 - 에어버스 등 항공사와 관련 업체, 그리고 Eclipse Foundation이 안전과 관련된 산업 장비에 대한 오픈소스 기반 개발을 위해 만든 단체
- Android Community
 - 주요 모바일 운영체제인 Google Android 운영체제를 지원하기 위한 단체



Linux

- GNU에서 관리하고 있는 오픈소스 운영체제
- X86 기반에서 벗어나 임베디드 기기, 모바일 등 다양한 하드웨어에 포팅
- 데스크탑이나 웹서버 위주에서 현재는 클라우드, 빅데이터 등의 분야에서 사실상의 표준 운영체제로 자리 잡음
- 모든 안드로이드 스마트폰 단말(한국 90.1%)
- 전세계 슈퍼 컴퓨터의 93.8%(2013년 11월 기준)
- Google, Twitter, Facebook, Amazon 등
- 개발자 8,000명, 회사 800개, 1,500만줄의 코드
- 3~6개월 주기의 커널 업그레이드

〈서버용 OS 점유율, 2011〉

이름	점유율
Red Hat	28.6%
Debian	12.2%
CentOS	10.6%
Fedora	9.5%
Ubuntu	8.7%
SuSE	11.6%
Others	18.6%



debian

fedora^f



CentOS



ubuntu





● MySQL(MariaDB)

- 전세계적으로 가장 많이 사용되고 있는 오픈소스 데이터베이스의 하나
- 페이스북 등 거의 대부분의 웹 서비스 업체에서 운영 데이터 저장용으로 사용
- 현재는 클러스터링, 분산(샤딩), 복제 및 백업 등 엔터프라이즈 환경에서 필요로 하는 고급 기능까지 지원
- 오라클이 인수(10억 달러) 후 "오픈코어 전략" 표방
- 오픈소스 결과물의 확장 기능이나 엔터프라이즈 기능 공개하지 않고 있음
- 이에 반발한 마이클 몬티가 막내 딸 이름을 붙여 MariaDB 시작



● Apache HTTP Server

- Apache 재단의 대표 프로젝트로 전 세계적으로 가장 많이 사용(46% 점유율)하는 웹서버
- NCSA의 HTTPd 코드를 기반으로 Linux에서 사용할 수 있도록 웹 서버 코드를 재 작성
- 커뮤니티 그룹과 미 델라웨어사와 합작하여 Apache Software Foundation 설립 (1999)

성공한 오픈소스 프로젝트

● Hadoop

- 빅데이터 분야의 대표적인 오픈소스 프로젝트
- 거의 모든 빅데이터 업체들이 사용하고 있는 기반 플랫폼
 - (예) Hortonworks, Cloudera, MapR, IBM, Oracle
- 빅데이터 관련 소프트웨어들의 생태계 역할
 - (예) Pig, Hive, HBase
- 모바일, 사물인터넷에 기반한 급속한 데이터 증가에 따라 더욱 중요도 증가



● Openstack

- 클라우드 업체인 Rackspace와 NASA가 협력하여 만든 오픈소스 클라우드 운영 시스템
- 클라우드 시스템 구축에 필요한 컴퓨팅, 네트워크, 스토리지 자원에 대한 가상화 및 운영 관리는 물론이고 보안, 워크플로우 등 거의 모든 요구사항 지원하는 사실상의 표준
- 클라우드 업체는 물론 가상화, 하드웨어, 네트워크, 스토리지 등 다양한 벤더들이 참여 중



● MongoDB

- 기존 RDB와는 다른 NoSQL 기반의 데이터베이스
- 빅데이터 시대의 도래와 함께 비정형 데이터를 대량으로 처리할 수 있는 장점 부각
- RDB와 유사한 인터페이스, 폭 넓은 사용자 저변, 높은 성능으로 NoSQL 분야에서 사실상의 업계 표준



- Pentaho
 - BI(Business Intelligence) 및 데이터 통합(ETL 등) 분야의 오픈소스 강자
 - Hadoop과 NoSQL 지원을 통해 빅데이터 시장 진입
 - 자본 유치와 고객 확보에도 성공
- PostgreSQL(Enterprise DB)
 - MySQL(MariaDB)과 함께 주요한 오픈소스 RDB 프로젝트의 하나
 - Enterprise DB를 통해 상용화된 버전 제공
 - Oracle DB와 뛰어난 호환성
- Wordpress
 - 오픈소스 블로그 플랫폼이자 WCM(Web Contents Management) 플랫폼
 - 전 세계 웹사이트의 22%, WCM 시장의 60% 이상 점유
- WSO2
 - Apache 라이선스 기반으로 한 오픈소스 미들웨어 스위트
 - ESB, BPM, BRM, API Manager, CEP 등의 컴포넌트



Why Open Source?

개방형 기술의 필요성 – 우리나라 IT 현실

- 특정 벤더의 운영체제와 브라우저에 대한 높은 종속성으로 인한 국가적인 보안 위기 초래
- 종속성을 낮추기 위한 방안으로 자유로운 사용과 수정이 가능한 오픈소스 부각
- 실제 다양한 국가와 기관에서 오픈소스를 사용한 성공 사례
 - 독일(뮌헨 시청 Ubuntu Linux 도입), 중국(캐노니컬과 공동으로 PC 용 Ubuntu 개발)

미래부, IE구버전 기술지원 종료에 대응...비상대응반 가동

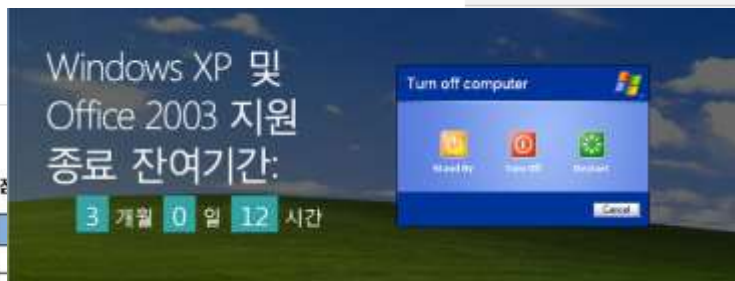
최종수정: 2016.01.12 16:45 | 기사입력: 2016.01.12 16:45

T + - ≡ ≡ ≡ ≡ ≡

< 지원종단 웹브라우저 >

구분	IE10	IE9	IE8	IE7	IE6	IE5
윈도우 XP	0.00	0.00	7.24	0.17	0.00	0.04
윈도우 VISTA	0.00	0.33	0.14	0.00	0.00	0.00
윈도우 7	8.02	11.98	7.44	1.66	0.00	29.1
윈도우 8	21.08	0.00	0.00	0.16	0.00	21.24
합계	29.1	12.31	14.82	2.41	0.83	59.47

(단위:%, '15년 하반기 국내 인터넷 이용환경 현황 조사)



IE 구버전 지원 중단, 버전 업그레이드 필수

입력날짜 : 2016-01-13 09:34

스크랩 프린트하기 목록



최신버전 업그레이드 필요
비상대응반 운영

플로러(IE) 구버전 사용자들이 보안위협에 노출될 위험에 빠
버전 제품에 대한 기술 지원을 2016년 1월 13일 02:00(한국

이에 따라 미래창조과학부(이하 미래부)는 윈도우7 및 윈도우8 OS 이용자들(50.34%)은 IE 11 업그레이드를, 윈도우XP 이용자들은(8.49%) 보안 지원이 가능한 새로운 운영체제를 설치해 보안을 강화해야 한다고 당부했다.

왜 오픈소스인가?

- 대내외 비즈니스 환경의 변화
 - 빠르게 변하는 비즈니스 주변 환경
 - "어떤 IT 기술" 보다는 "필요한 비즈니스"를 "빠르게" 하는 것이 중요한 시대
 - Mobile, SNS 혁명
 - 대량 데이터 분석의 필요성

방금 1분 동안 !!



293,000개의 상태 업데이트
20G 이상의 로그 데이터



10만개의 트윗 메시지



2억 개의 이메일이 발송

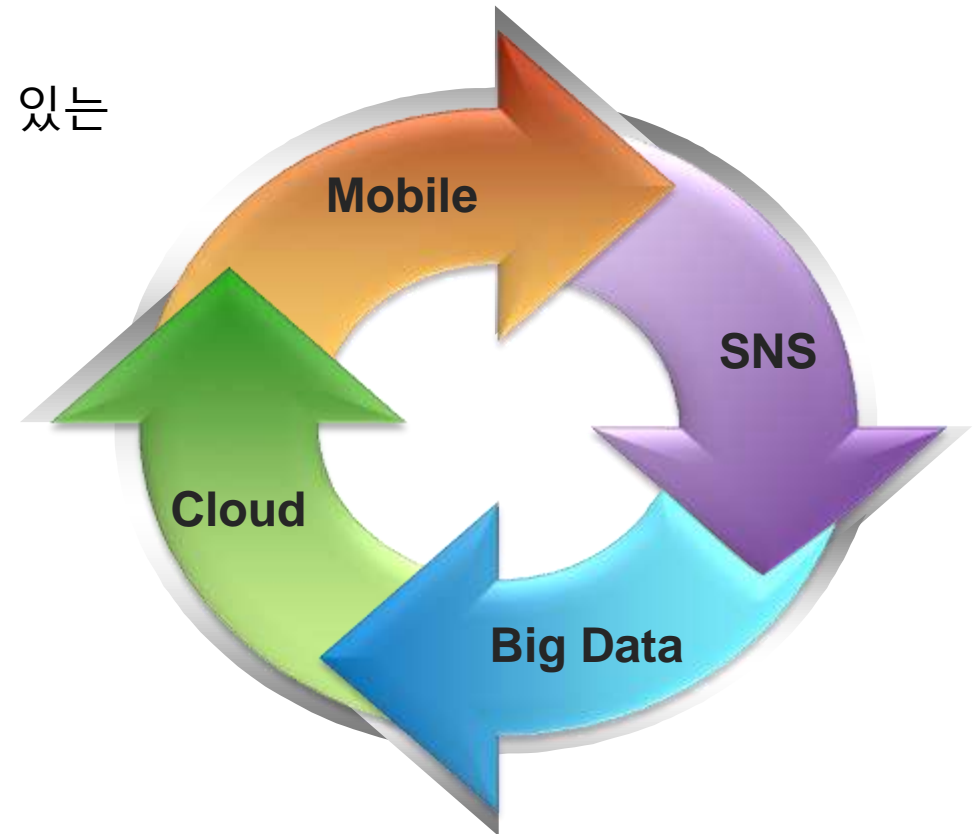
왜 오픈소스인가?

- 비즈니스 환경의 변화에 맞춘 혁신적 IT 기술의 필요성
 - Mobile, SNS의 발달로 트래픽 증가
 - 과거에는 주요 기술들을 소수의 IT 벤더에게서 제공 받아야 했음
 - 과거의 IT 기술로 커버하기 힘든 상황
 - 새로운 요구사항을 빠르게 수용할 수 있는 기술의 필요성



왜 오픈소스인가?

- 빅데이터, 클라우드(가상화) 등 혁신적 신기술
 - Mobile, SNS에서 발생하는 대용량 데이터를 처리할 수 빅데이터 기술 적용
 - 서비스 수요에 맞추어 빠르게 인프라를 구축할 수 있는 클라우드, 가상화 기술 적용
 - 빅데이터, 클라우드는 현재도 빠르게 발전하고 있는 Emerging Technology
 - 빠른 기술 발전을 수용하면서도 다수의 참여자가 평등하게 사용할 수 있는 기술 기반의 필요성



왜 오픈소스인가?

- 다수의 참여자 모델
 - 소수의 개발자들에 의해 비밀스럽게 관리되는 상용 소프트웨어보다 다수의 개발자/사용자가 참여하는 오픈소스가 기술 발전 속도가 월등히 빠름
 - (예) 하드웨어 벤더들이 독점적으로 개발하던 운영체제가 Linux 기반으로 변경
- 독립적 플랫폼의 필요성
 - 시장 영향력이 큰 플랫폼이 특정 벤더에 종속적이 되는 것을 방지
 - (예) Allseen, XENIVI, Polarsys
 - (예) PC 운영체제를 독점한 특정 기업에 의해 응용 소프트웨어 시장도 독과점 발생
- 혁신적 기술의 발전에 유리
 - 클라우드, 빅데이터, 사물인터넷 등처럼 발전 속도가 빠른 최신 기술에 대한 구현이 용이 - 정보나 소스에 대한 공유가 자유로움
 - 표준이 같이 발전하는 경우 그것의 구현체로서 오픈소스가 최적의 역할
 - (예) Openstack, Hadoop



기업과 오픈소스

IT 트렌드 – '불편한 현실' 2016년 IT 전망 5가지

- 레거시 벤더들이 직면한 암울한 미래
 - IDC는 보고서를 통해 "향후 5년 내 현존하는 IT 벤더들 가운데 30%가 사라지게 될 것이다"라는 전망
 - 현재도 일어나고 있는 상황: HP와 Citrix의 분사, Dell과 EMC 합병
- 클라우드 공급업체들도 걸러질 것이다
 - 포레스터 리서치는 '이제 클라우드 업체를 추려 낼 시점'이라고 진단
- 더욱 커지는 빅데이터의 역할
 - IDC에 따르면, 빅데이터는 이제 막 시작 단계로 현존하는 앱 가운데 1%만 인지 서비스를 사용 중. 3년 후, 2018년이면 이 숫자는 50%로 증가
 - 빅데이터가 넘어야 할 문제는 스토리지. 기계 학습 서비스에 투자할 수 없는 소규모 클라우드 서비스들은 도태
- 소프트웨어 기업으로 변해가는 기업들
 - IDC는, "2018년이면 DX 전략을 취한 기업들이 소프트웨어 개발에 있어 그렇지 못한 기업들보다 두 배의 역량 갖게 될 것이다"
- 개발자가 부족해진다
 - 기업 IT 부문과 IT 벤더 모두 비즈니스의 근간을 이루는 IT 솔루션을 추구
 - 둘 사이의 차이점이 줄어들 것이라는 점
 - 한정된 인재를 둘러싼 기업 IT 부문과 IT 기업들의 경쟁은 앞으로 더욱 치열하게 전개

오픈소스 기업과 다른 IT 벤더의 주가 비교

Red Hat, Inc. (RHT) ★ Watchlist
63.54 +1.94(+3.15%) NYSE - As of 4:02PM EST

Beat the market

Get the app



국내외 오픈소스 시장

- IDC 보고서에 의하면, 2015년 전세계 오픈소스 SW 매출은 619억 달러
- 연평균 18.8%의 성장률로, 2017년에는 900억 달러에 이를 것으로 전망
- 전세계적으로 오픈소스 시장 급성장 하고 있으며, 영국, 호주, 러시아 등의 국가에서는 정책적으로 오픈소스 활성화 정책
- 국내 OSS 시장은 2009년부터 급성장하여 2015년 618억원 규모 시장

전 세계 오픈소스 시장



국내 오픈소스 시장



기업들이 오픈소스를 통해 해결하고자 하는 당면 과제
비용 절감 68%, IT 인프라 환경 개선 62%
51% 기업이 신규 인력 채용에 오픈소스 사용이 도움

Source: 2014 Future of Open Source Survey Results

- OSI(Open Source Initiative)에서 제시하는 오픈소스가 비즈니스에 주는 이익
 - 경쟁 업체 대비 빠른 비즈니스 서비스 구현과 높은 품질을 통한 차별성
 - 개방형 표준 기반의 실제 동작하는 소프트웨어 확보
 - 훨씬 빠른 버그 픽스(bug fix)와 기능 추가, 그리고 높은 품질과 안정성
 - 낮은 TCO(저 비용)
 - 원하는 제품과 기술에 대한 협력의 장(생태계) 확보
 - 높은 보안(공개된 소스를 통해 보안 문제 빠르게 해결)
 - 중복 투자를 방지하고 표준화된 도입, 운영 절차를 통해 효율성 향상
 - 기존 직원들에게 동기부여 및 신규 직원 충원 용이성
 - 인프라를 Commodity화 하고 이에 대한 지원 연속성 확보

상용 소프트웨어와 오픈소스의 비교

구분	상용 SW	오픈 소스
비용분석	<ul style="list-style-type: none"> 초기 도입비용이 높음 유지보수 비용 및 시스템 개선비용 높음 	<ul style="list-style-type: none"> 초기 도입비용이 낮음 유지보수 비용 및 기능 추가 비용이 낮음
제약사항	<ul style="list-style-type: none"> 라이선스 계약으로 인한 제약사항 (사용료 지급) 	<ul style="list-style-type: none"> 사용상의 제약사항 없음 (무료)
배포형태	<ul style="list-style-type: none"> 라이선스 계약으로 인한 바이너리 제공 소스코드는 기업비밀로 유지함 	<ul style="list-style-type: none"> 소스코드가 공개 라이선스 조건하에 자유롭게 배포,수정
성능분석	<ul style="list-style-type: none"> 비교적 큰 시스템 환경에서의 높은 성능 나타냄 	<ul style="list-style-type: none"> 다양한 환경에서 최적화된 설정으로 높은 성능치를 나타냄 (Intel, PPC, s390등)
기술성	<ul style="list-style-type: none"> 문제점 발생시 폐쇄적인 운영으로 취약점 보유 	<ul style="list-style-type: none"> 소스코드의 공개로 빠른 문제점 해결 유지보수 및 업그레이드 용이, 독점피해 방지
확장성	<ul style="list-style-type: none"> 시스템 환경에 따라 호환성은 보장 높은 적용비용을 지불해야 함 제한된 시스템 운영환경에서 용이 	<ul style="list-style-type: none"> 소프트웨어간 적용비용이 낮음 기능추가 비용이 낮음
공급권	<ul style="list-style-type: none"> 최초도입 개발업체 또는 벤더에게 공급 개발업체 문제시 도입 고객에게도 심각한 영향을 미침 	<ul style="list-style-type: none"> 동일 솔루션에 대한 다수의 업체로 부터 지원 및 공급이 가능, 사용자의 선택권이 넓음

- 클라우드 기반환경을 플랫폼 형태로 만들어 재판매 할 경우 강력한 원가절감, 솔루션 유연성을 가짐(예: 고객 유지보수 비용으로 포함)

- 오픈소스는 비즈니스(수익) 모델인가?
 - No. 자체적으로 시장을 형성할 수 없는 “비즈니스 전략”
 - 라이선스 전략에 기반한 소프트웨어 개발 및 배포 모델
 - 수익 모델을 만들거나 상용화에 어려움
 - 간접적 가치 생산 사례: 킨들 무료 배포 후 책을 판매하는 방식
- 구글(또는 페이스북)의 오픈소스 기반 서비스 모델
 - 오픈소스를 활용하여 회사의 가치 향상
 - 우수한 조직은 자신들의 데이터로부터 어떻게 가치를 뽑아내는지 알고 있다
 - 그렇다면, 어떻게 데이터를 뽑아낼 수 있는가? 소프트웨어!!!
 - 소프트웨어는 누가 만드는가? 프로그래머들 !!!
 - 프로그래머들이 가장 효과적으로 일할 수 있는 툴이 무엇인가?
 - 그것이 오픈소스(기술)이고, 그것을 "사용"하여 "서비스"를 만들고, 그것을 통해 "가치"를 만들어낸다

● 오픈소스에 가치를 부여하는 방식

- 오픈소스로 간접적 가치 생산 모델
 - 듀얼 라이선싱: MySQL, MongoDB처럼 GPL 버전은 무료, Non-GPL 버전은 유료
 - 코어 오픈 모델: 기본 기능은 무료로 하고, 고급 기능(예: 클러스터링)을 유료화
- 오픈소스 전문 컨설팅 회사: RedHat, Pantaho 등처럼 전문가로서 커뮤니티에서 만들어진 오픈소스에 대한 사용 방법을 컨설팅하고, 제품처럼 패키징 하고, 브랜드를 붙이고, 문서를 만들고, 교육을 하는 과정을 유료화
- 오픈소스 기반으로 상용 제품을 제작: 일반적인 상용 소프트웨어와 동일하며, 다만 그 기반으로 오픈소스를 사용한다는 차이
- PaaS/SaaS 모델: 오픈소스의 전문가로서 해당 컴포넌트 기반 서비스를 만들어 그 것을 유료로 판매
- HW 결합 모델: 하드웨어와 오픈소스 소프트웨어를 결합(어플라이언스)하여 유료화

기업의 오픈소스 활용 전략

● 오픈소스를 활용한 전략의 사례

- 마케팅 활용 – 오픈소스 공개 시 관심 유도를 통한 마케팅 효과
- 지원 비용 절감 – 사용자가 직접 트러블슈팅에 참여
- 개발 비용 절감 – 코드 공개 시 엔지니어링 작업 참여 인력 증가
- 경쟁 우위 확보 – 오픈소스화를 통해 시장 진입을 유리하게 하거나, 반대로 사실상의 표준인 오픈소스 플랫폼 전략으로 시장에서의 우위 확보
(예) HTML5 표준에 Flash 기술을 공개한 Adobe
- 개방형 표준 수립 – 기술 혁신을 주도하는 과정에 오픈소스 기반의 개방형 표준 활용
(예) 자사의 오픈소스를 기증 해 차량용 Infotainment 컨소시엄 구축

● 오픈소스 재단에 대한 기업의 지원 전략

- 오픈소스 기술에 대한 지분 확보
- 부가적으로 신뢰성(마케팅 효과), 친근한 이미지(인재 확보에 도움) 확보

76%가 향후 2~3년 내에 가장 큰 영향을 받는 분야로 교육 분야를 지목

Source: 2014 Future of Open Source Survey Results

KERRIN PERNICIARO, DIRECTOR OF WEB & IT STRATEGY, SUNY STONY BROOK



"Many colleges and Universities are already using open-source solutions (OSS) and will continue do so in the future because of how quickly and cost-effectively OSS can help to deploy projects. Due to budget constraints in recent years, faculty and staff have been asked to do more with less, so it makes sense to seriously consider open-source alternatives when faced with making strategic decisions...In addition, many incoming students...expect it to be part of the curriculum, whether in the classroom or as part of their campus job."



- 이미 많은 대학교에서 오픈소스를 사용 중
- 예산 상의 제약으로 적은 비용으로 더 많은 성과를 내야 하는 상황
- 학생들에게도 오픈소스는 새로운 동기 부여



오픈소스 성공 사례

NIPA 공개SW 성공사례집

- 각 분야 별 오픈소스 적용 성공 사례
- 정보 시스템 분야
 - 넥슨: IT 시스템 관리/모니터링/운영
- 모바일 분야
 - 한양대학교: 모바일 스마트 캠퍼스
- 클라우드&빅데이터 분야
 - GS샵: 오픈소스 기반 상품 추천 시스템
- 임베디드 분야
 - 핸드스튜디오: 오픈소스 기반 스마트 TV 앱

기관	개발 프로젝트	활용 공개SW	분야	페이지
GS샵	상품 추천 서비스 플랫폼	하둡 에코시스템(Hadoop ECO system), 머아웃(Mahout), 물류(Flume), 몽고DB(MongoDB), 클루몬(Clumon)	클라우드&빅데이터	P. 52
HP	물류스 네트워크	오픈플로우(OpenFlow)	정보시스템	P. 08
KDB대우증권	비즈니스 프로세스 관리 시스템	유엔진 BPM(uEngine BPM)	정보시스템	P. 32
KTH	빅데이터 플랫폼 'DAISY(Data Intelligence System)'	에스퍼(Esper), 루센(Lucene), 하둡(Hadoop), 차트(Charl), R, 스크(Sqoop), 하이브(Hive), 피그(Pig), 머아웃(Mahout), 카프카(Kafka)	클라우드&빅데이터	P. 54
LG유플러스	사용자 패턴 분석 시스템	맵리듀스(MapReduce), 몽고DB(MongoDB)	정보시스템	P. 18
네오위즈	대용량 카백션 풀	SQL 릴레이(SQL Relay), 리눅스(Linux), 센트OS(CentOS), 아파치 웹서버(Apach HTTP Server), 마이SQL(MySQL), 포스트그레SQL(PostgreSQL), 루비(Ruby), 톰캣(Tomcat), 자바(Java), PHP, IRCd	정보시스템	P. 12
네이버	N클라우드	오픈플로우(OpenFlow)	클라우드&빅데이터	P. 50
넥슨	IT 인프라 운영 관리 시스템	CACTI, 라눅스, 마이SQL(MySQL), 몽고DB(MongoDB), 카산드라(Cassandra), 멤베이스(Membase), 멤캐시드(Memcached), 바니시(Varnish)	정보시스템	P. 14
로보티즈	다윈-오피(DARwin-OP)	우분투 데스크톱 9.10(Ubuntu Desktop 9.10), 클론질라(Clonezilla)	임베디드	P. 60
산업통상자원부	클라우드 시범 풀	오픈젠(OpenJen), 유엔진 B-PaaS(Engine B-PaaS), KVM, 큐브리드(Cubrid), HAProxy, Git, 서브버전(Subversion), 아파치 웹서버(Apach HTTP Server), 제이보스(JBoss), 펄스토크(Fastack), 전자정부 표준 프레임워크(eGovFrame)	클라우드&빅데이터	P. 56
서울대학교 중앙도서관	기관 레보지토리 'S-Space'	디스페이스(Dspace), 센트OS(CentOS), 아파치 HTTP 서버(Apach HTTP Server), 톰캣(Tomcat), 메이븐(Maven)	정보시스템	P. 36
서울시	서울 열린 데이터 광장	릴파인더(RelFinder), 이그지비트 2.0(Exhibit 2.0), 루비(Ruby), 로드라이브(LodLive), 구글 리파인(Google Refine), 마이SQL(MySQL)	정보시스템	P. 26
서울시	온라인 콘텐츠 플랫폼	워드프레스(WordPress), 리눅스(Linux), 아파치 웹서버(Apach HTTP Server), 마이SQL(MySQL), PHP, 스프링 프레임워크(Spring Framework)	정보시스템	P. 10
순천시	2013순천만국제정원박람회 홈페이지와 모바일 웹	킨스큐 RB CMS(KimsQ RB CMS)	모바일	P. 44
안전행정부	전자정부 표준 프레임워크	스프링(Spring), 이클립스(Eclipse), 하이버네이트(Hibernate), 제이쿼리 모바일(Query Mobile) 등 45종	정보시스템	P. 16
에어부산	모바일 웹 앱 (m.airbusan.com)	제이쿼리 모바일(Query Mobile)	모바일	P. 40
엔씨소프트	게임 로그 데이터 분석	R, 하둡(Hadoop), 피그(Pig), 하이브(Hive), 캐스캐이딩(Cascading), SVN(Subversion Repository), 리눅스(Linux), 이클립스(Eclipse)	정보시스템	P. 24
전쟁기념관	전시안내 시스템	드루팔(Drupal)	정보시스템	P. 30
정보통신 산업진흥원	전자문서 유통서비스	아파치(Apach), 톰캣(Tomcat), 레드햇 엔터프라이즈 리눅스(RHEL), 큐브리드(Cubrid)	정보시스템	P. 22
한국고용정보원	차세대 종합 고용정보 서비스	전자정부 표준 프레임워크(eGovFrame), 폰갭(PhoneGap)	정보시스템	P. 34
한국발명진흥회	특허평가시스템 'SMART3'	아파치(Apach), 톰캣(Tomcat), 스프링 프레임워크(Spring Framework), 레드햇 엔터프라이즈 리눅스(RHEL)	정보시스템	P. 28
한국주택금융공사	스마트 주택금융 앱	폰갭(PhoneGap), 제이쿼리 모바일(Query Mobile)	모바일	P. 46
한양대학교	스마트 캠퍼스	제이쿼리(Query), 마이SQL(MySQL), 제이쿼리 모바일(Query Mobile), 스프링 3.0(Spring 3.0), 아파치(Apach), 몽고DB(MongoDB)	모바일	P. 42
핸드스튜디오	스마트TV 플랫폼	제이쿼리(Query), 마이SQL(MySQL), 마이바티스(Mybatis), 저지(Jersey), Ffmpeg, Log4j, 리눅스(Linux)	임베디드	P. 62
현대중공업	선박의 유체 해석	오픈폼(OpenFOAM)	정보시스템	P. 20



오픈소스 기반 아키텍처

오픈소스 기반 솔루션 아키텍처 전략

Ⅰ 전략 및 계획 부분은 상업용 소프트웨어 아키텍처를 따름

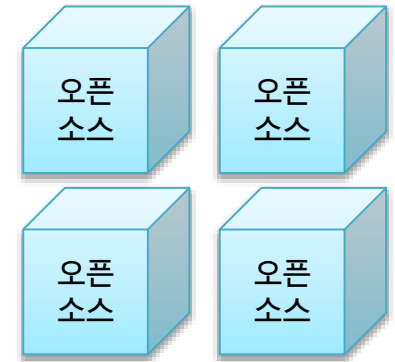
Ⅱ 기술 및 구현 부분은 오픈소스 소프트웨어의 아키텍처를 차용



사용자 및 시장의 요구 사항



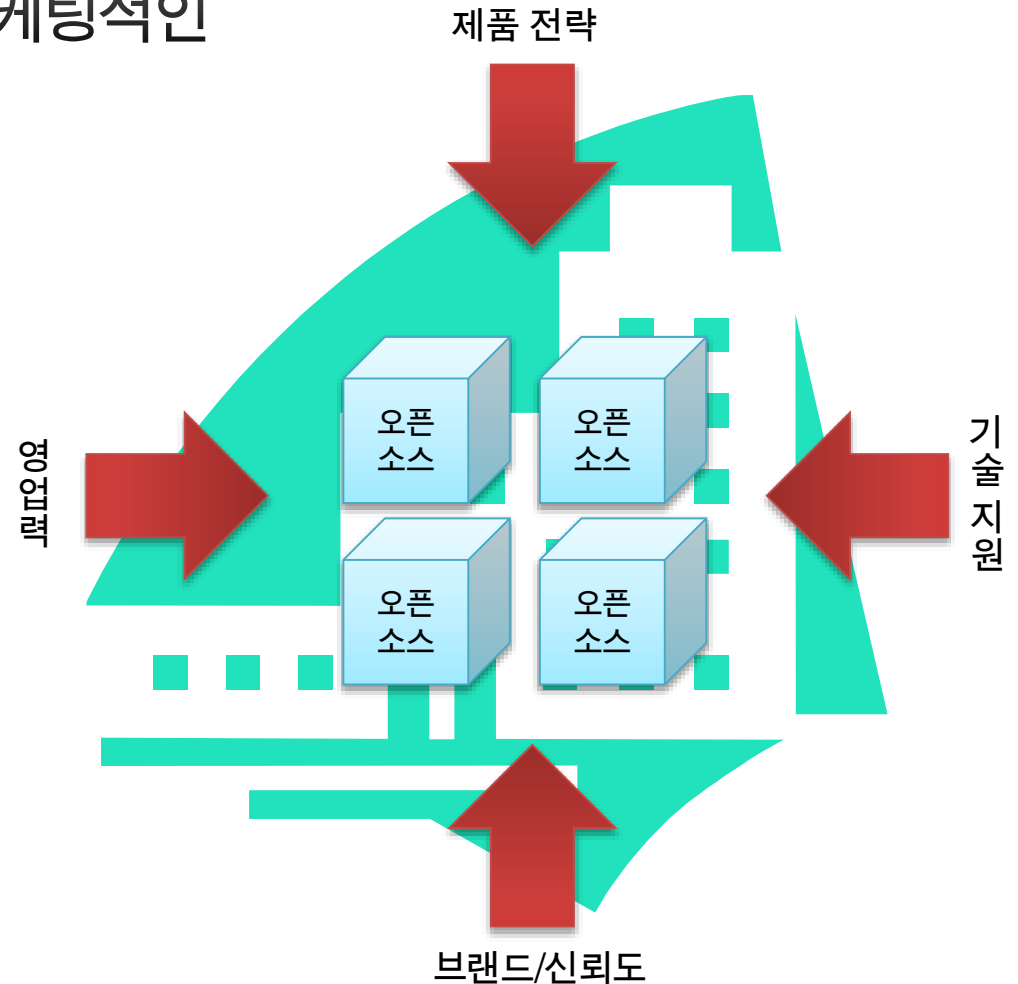
요구사항에 맞는 솔루션의 스펙을
정의하고 출시 일정, 타겟 고객
등을 결정



제품 구현에 필요한 기반 오픈소스
기술과 컴포넌트를 선택하고
요구사항에 맞게 통합(패키지화)

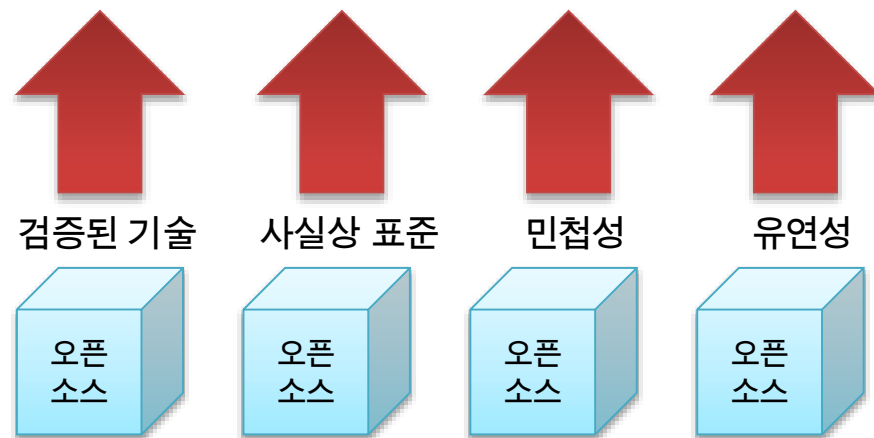
오픈소스 기반 솔루션 아키텍처 전략

| 오픈소스에 부족한 제품 전략, 마케팅적인 부분을 회사가 가진 역량(브랜드, 영업/기술력 등)을 통해 보완

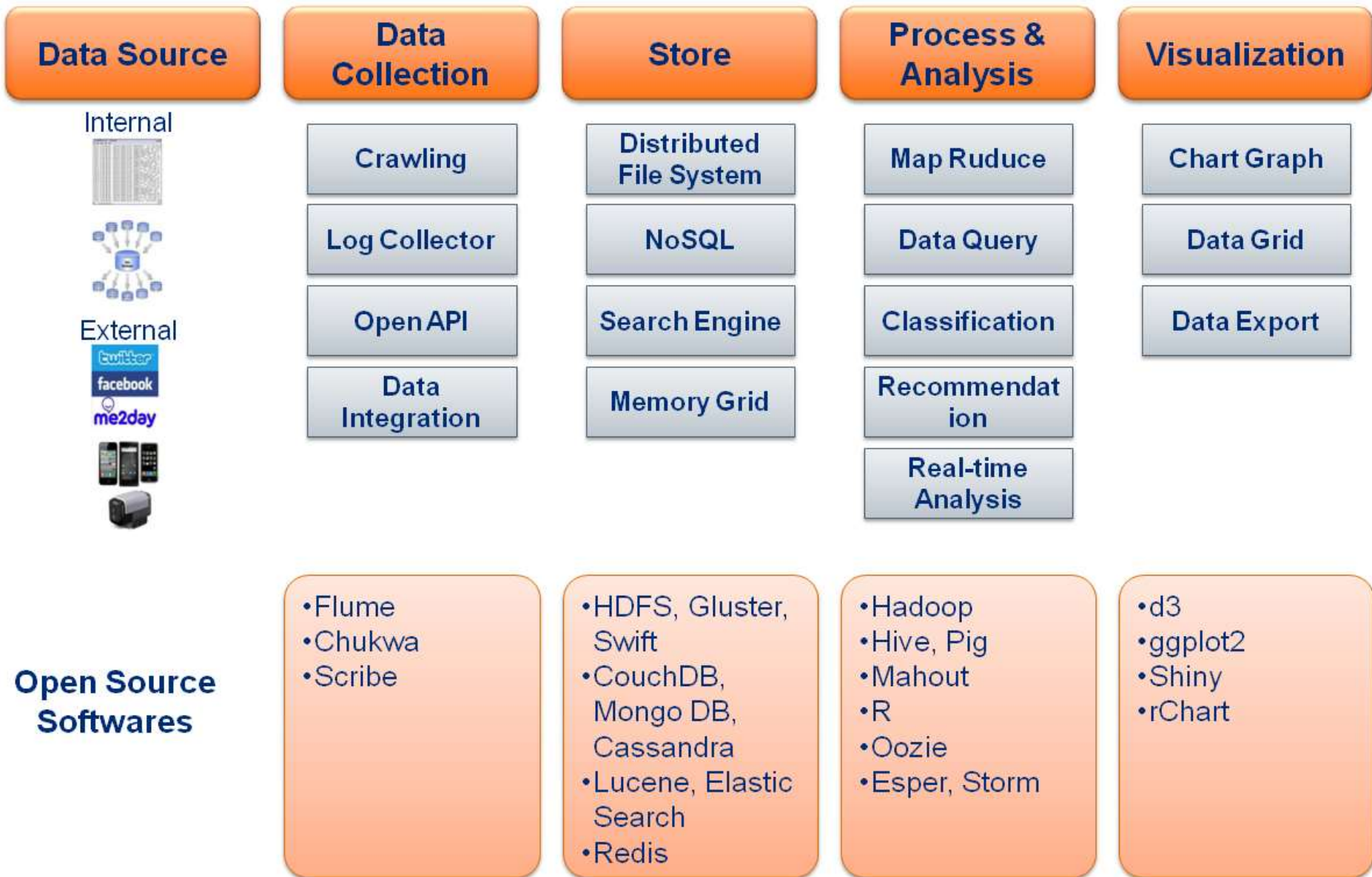


오픈소스 기반 솔루션 아키텍처 전략

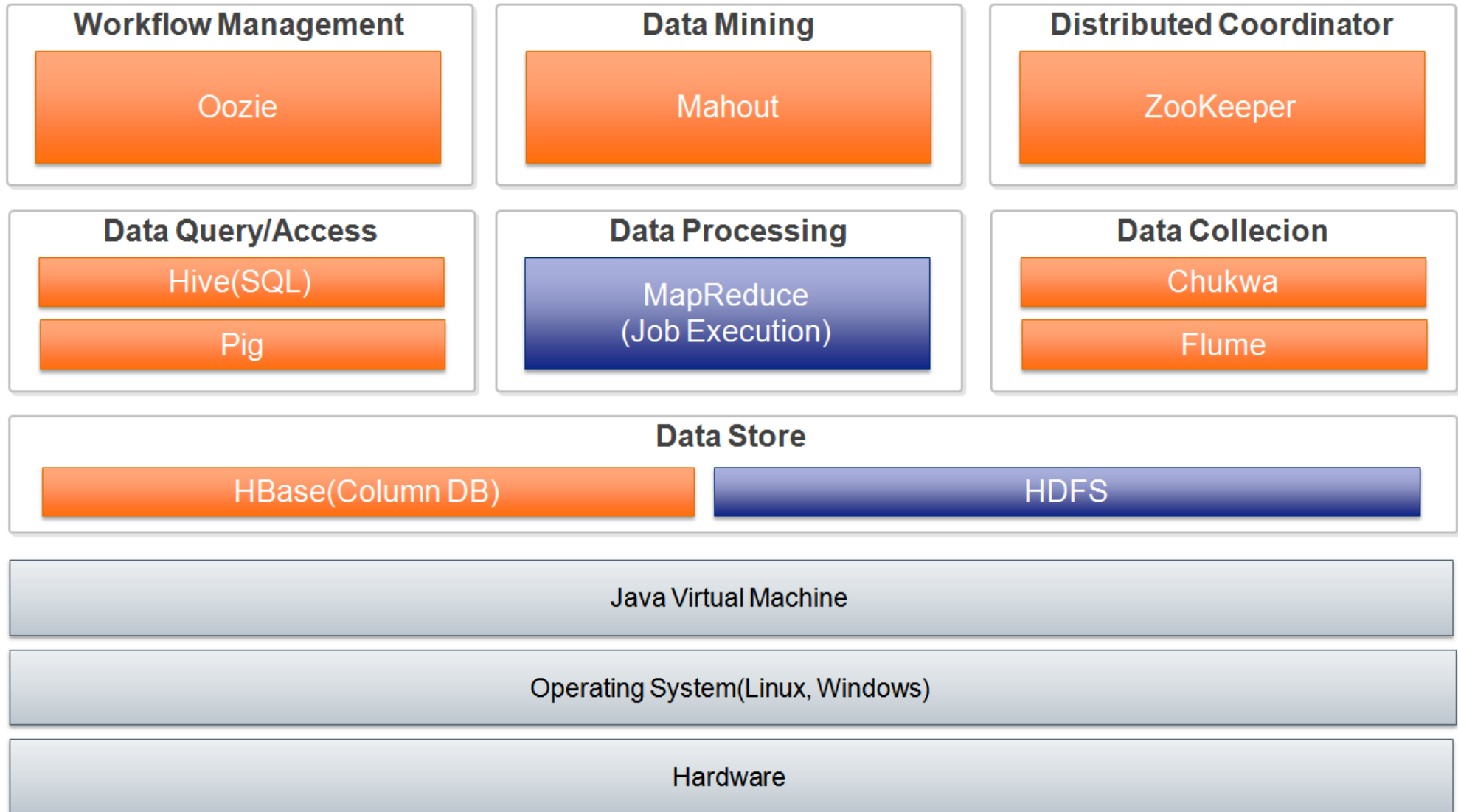
- 오픈소스를 통해 구현에 필요한 시간을 단축하고, 저변이 넓은 검증된 기술을 통해 솔루션의 기반을 견고하게 함
- 상용 솔루션이 필요로 하는 성능, 재사용성, 유연성 등을 쉽게 확보
- 전략에 맞는 아키텍처를 수립하고 그에 맞는 오픈소스 컴포넌트들을 선택하여 패키지화(통합)



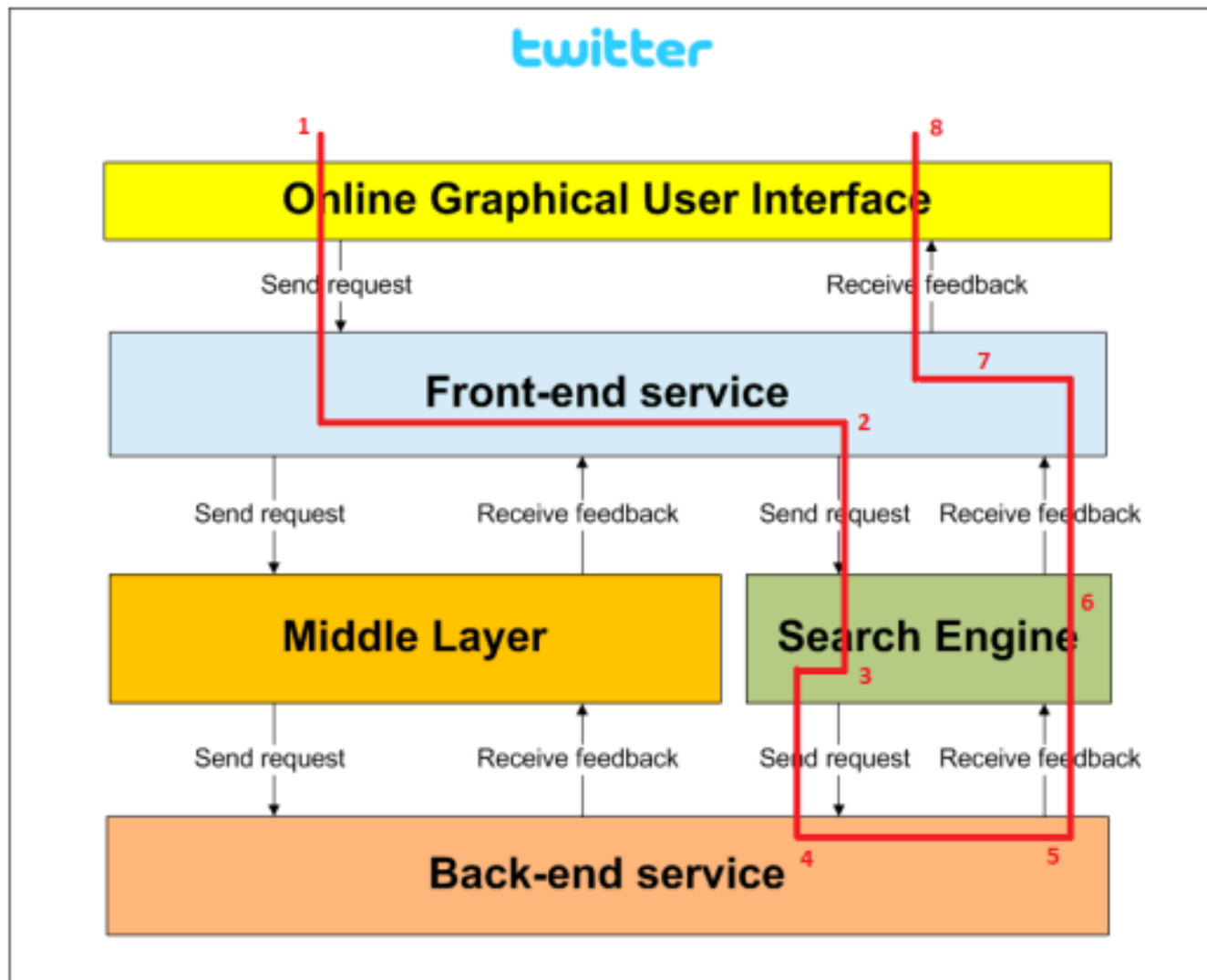
오픈소스 빅데이터 컴포넌트



Apahce Hadoop 오픈소스 생태계



Twitter 오픈소스 기반 아키텍처



Twitter 오픈소스 기반 아키텍처

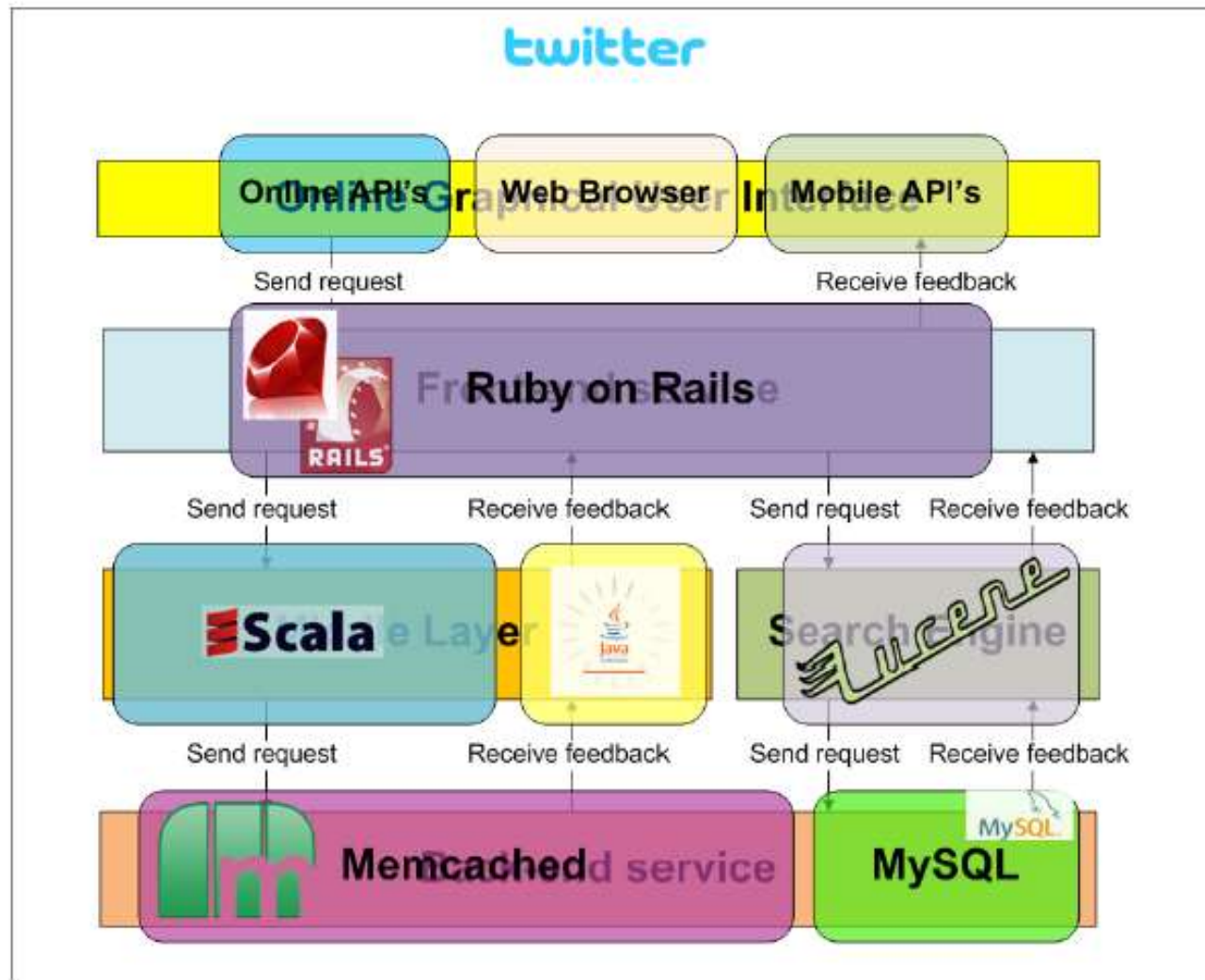
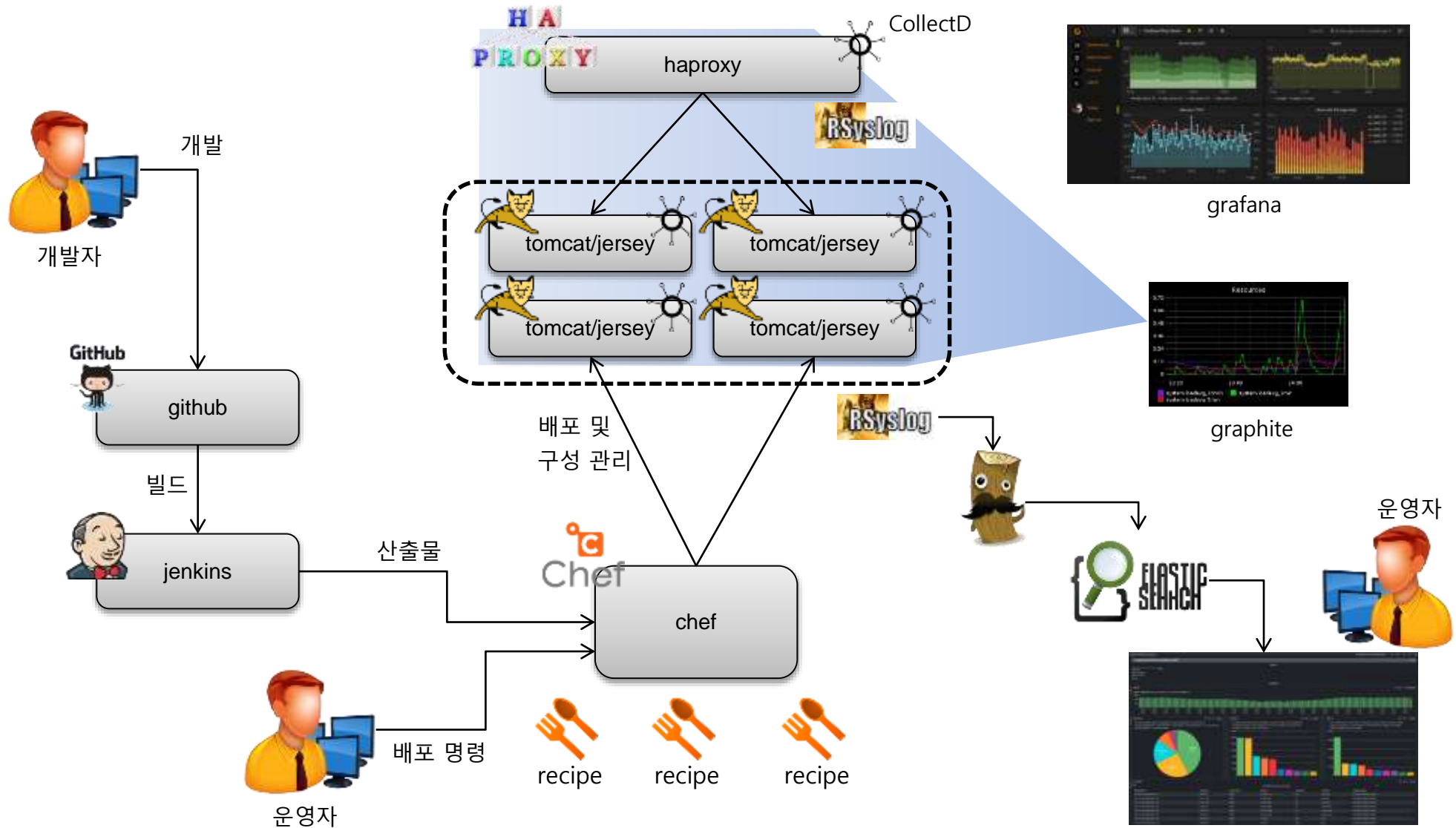


Figure 8: Software overlay

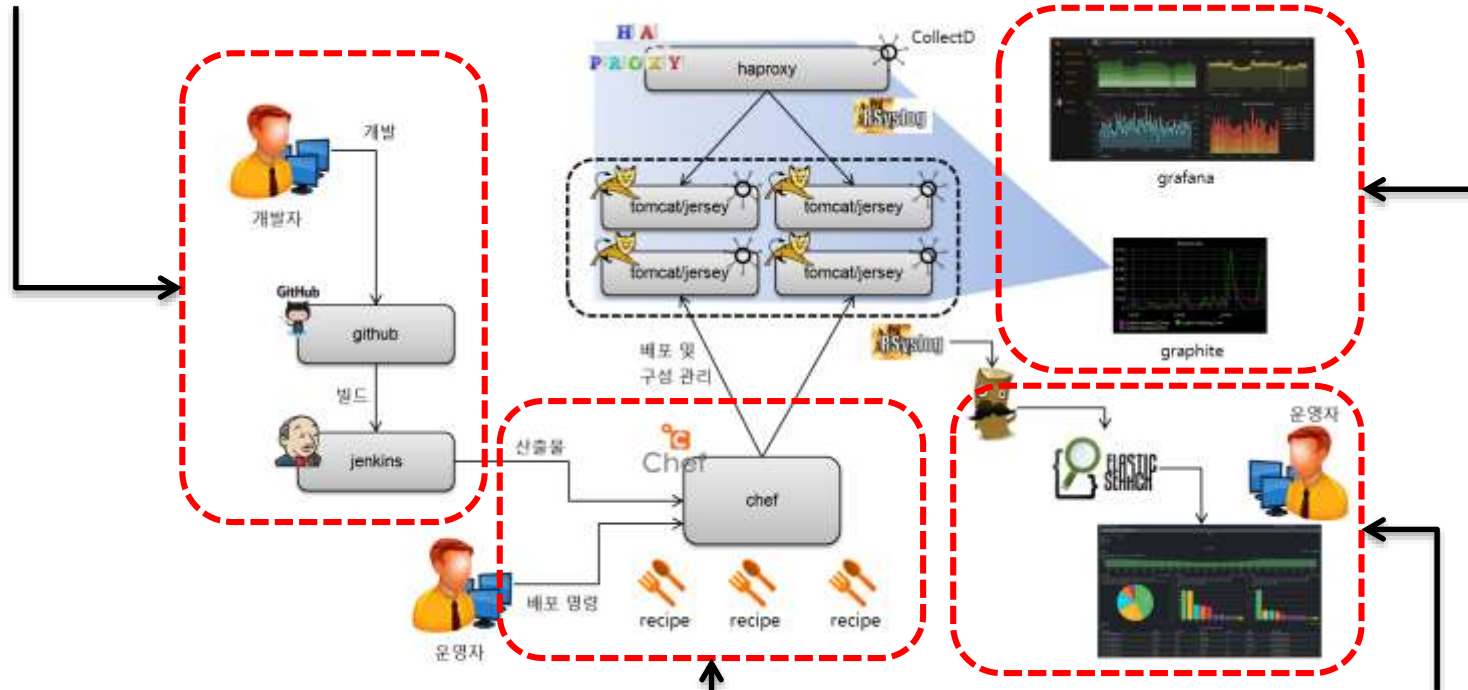
REST API 기반 PaaS Cloud 아키텍처



Lesson Learned

지속적인 빌드

문제: 개발 산출물을 클라우드에
반영하여 지속적인 업그레이드
해결: jenkins를 통해 개발 산출물
자동 빌드 및 반영



자동화된 Provisioning 및 구성 관리

문제: 신규 시스템 추가에 많은 시간 소요
해결: 신규 시스템 추가, 구성 변경, 새로운 개발
산출물 반영을 chef를 통해 관리

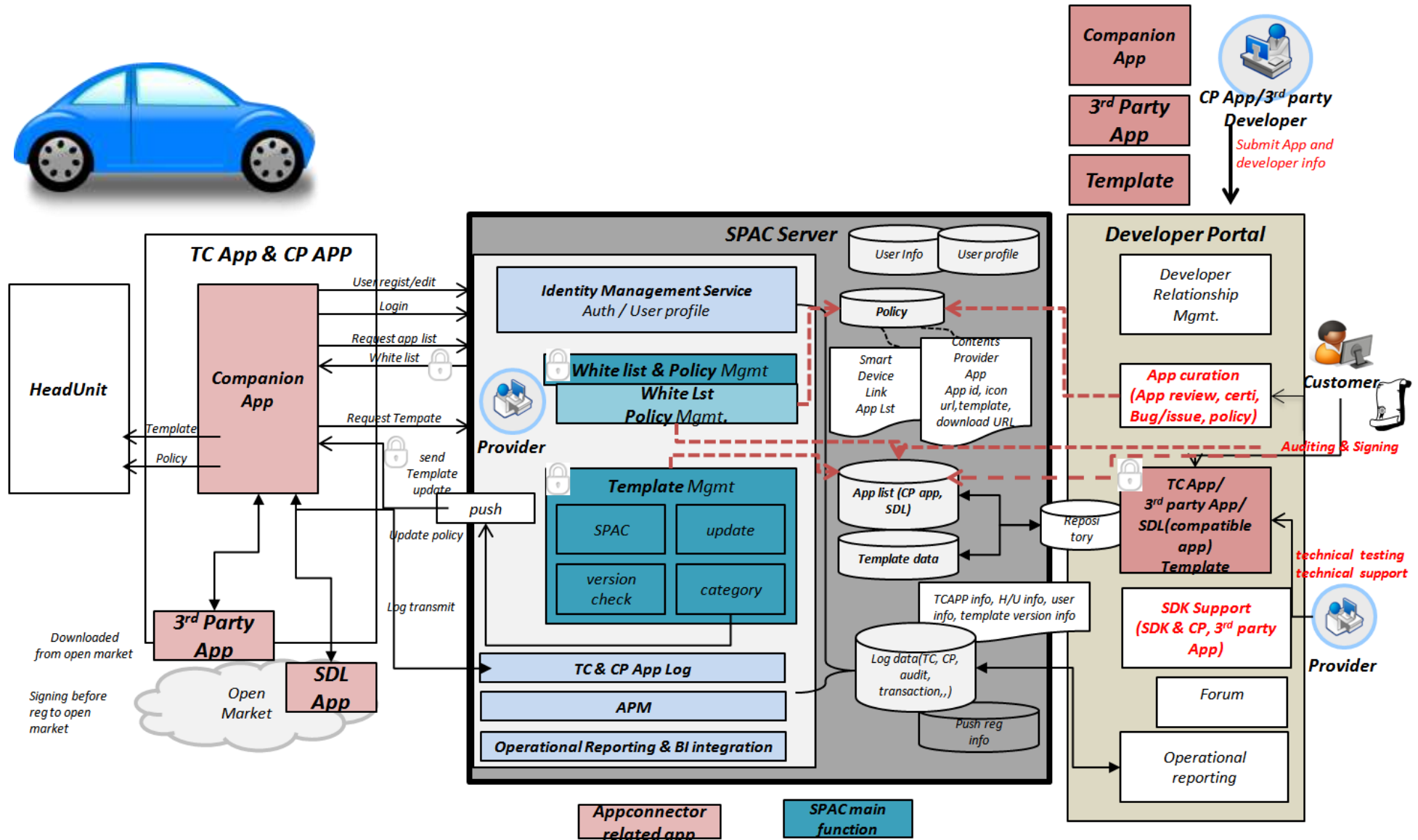
시스템 Health 모니터링

문제: 대규모 시스템에 대한 모니터링의
어려움
해결: collectd를 통한 개별 시스템
모니터링 정보 수집,
graphite/grafana를 통한 시각화

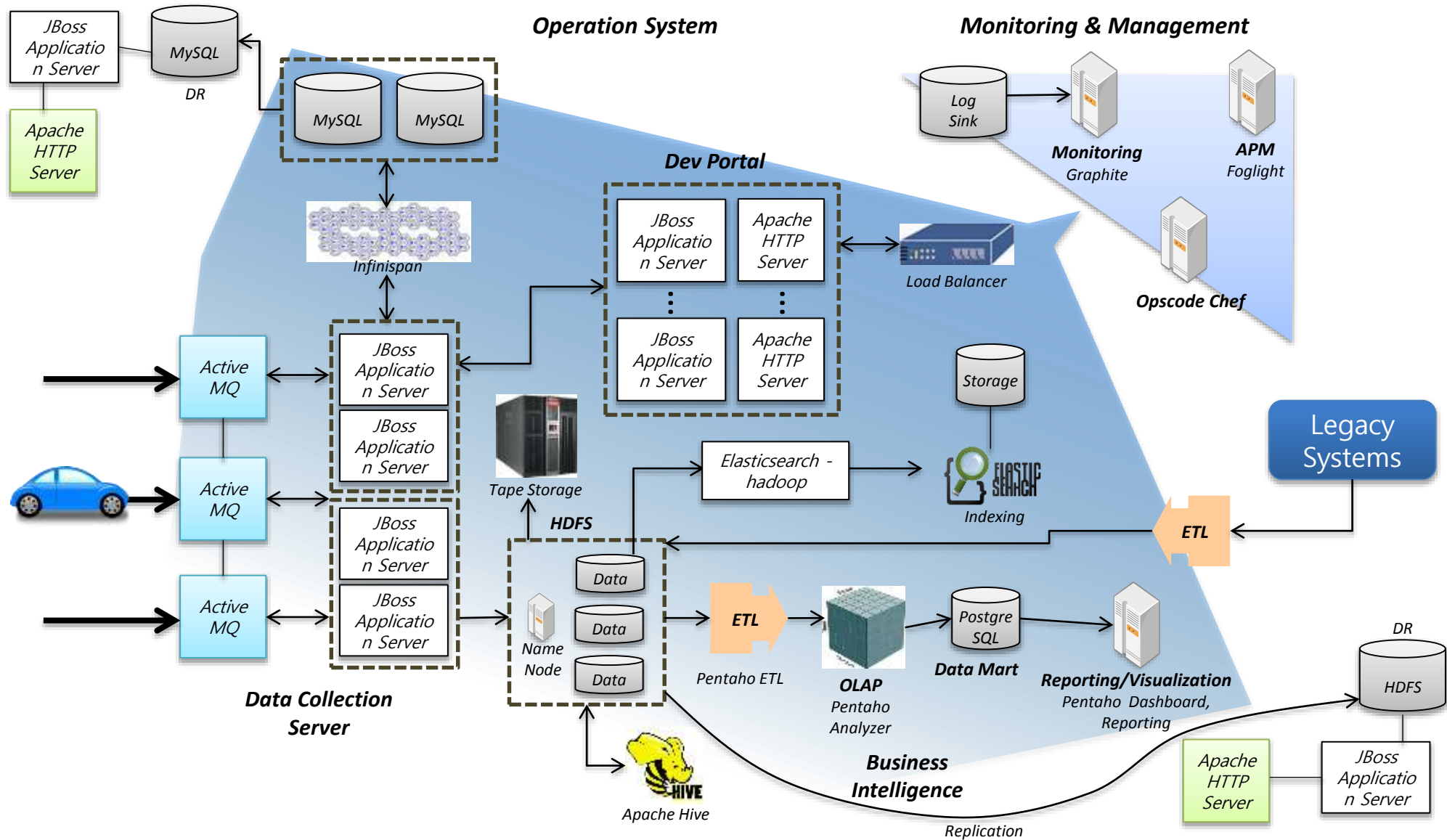
로그 수집 및 분석

문제: 장애 발생 시 산재된 각 시스템 로그 분석
어려움
해결: rsyslog + logstash 통해서 원격 로그를
중앙 수집하고, ElasticSearch를 통해
Indexing, Kibana를 통한 시각화

스마트카 – 사물인터넷



스마트카 – 사물인터넷

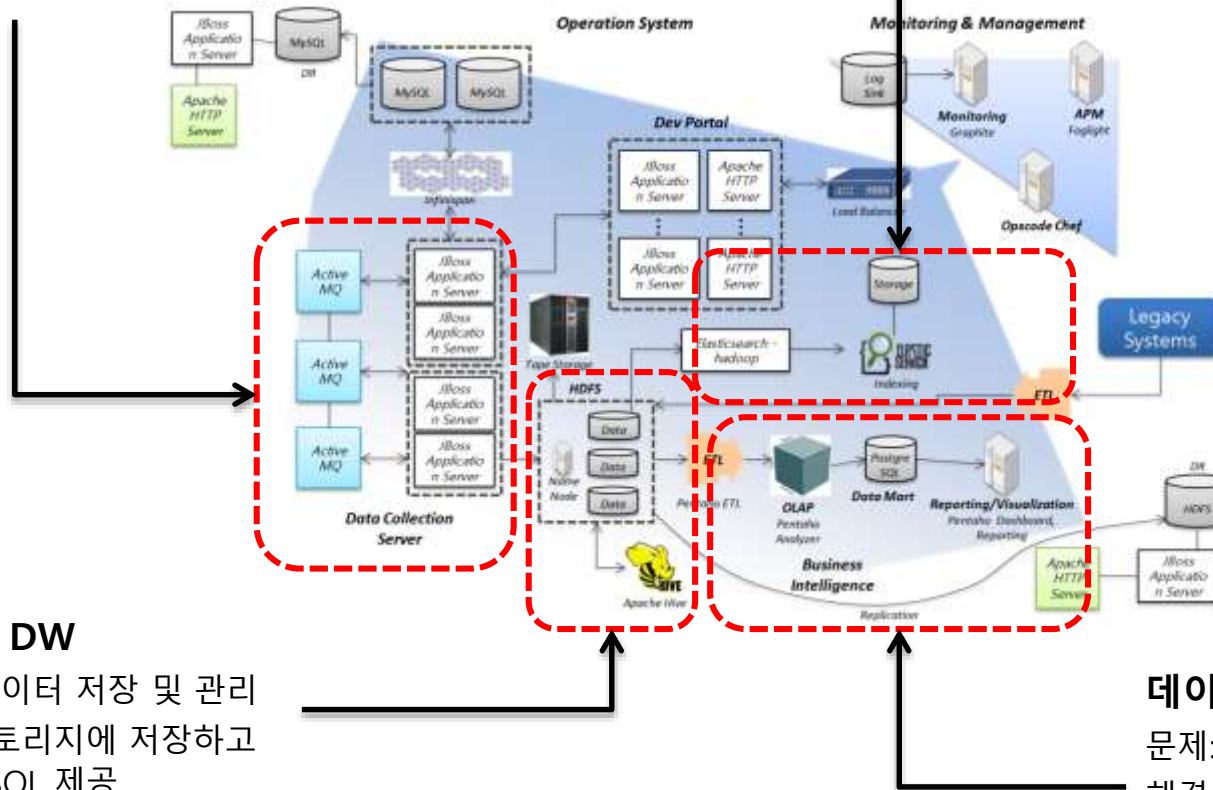


Lesson Learned

데이터 수집 서버

문제: 불안정한 네트워크로 인한 단말 제어, 데이터 수집의 어려움

해결: ActiveMQ를 통한 MQTT 통신 및 Jboss Application Server 클러스터링 통한 확장성



데이터 저장 및 DW

문제: 대용량 빅데이터 저장 및 관리
해결: Hadoop 스토리지에 저장하고 Hive를 통한 SQL 제공

Data Discovery

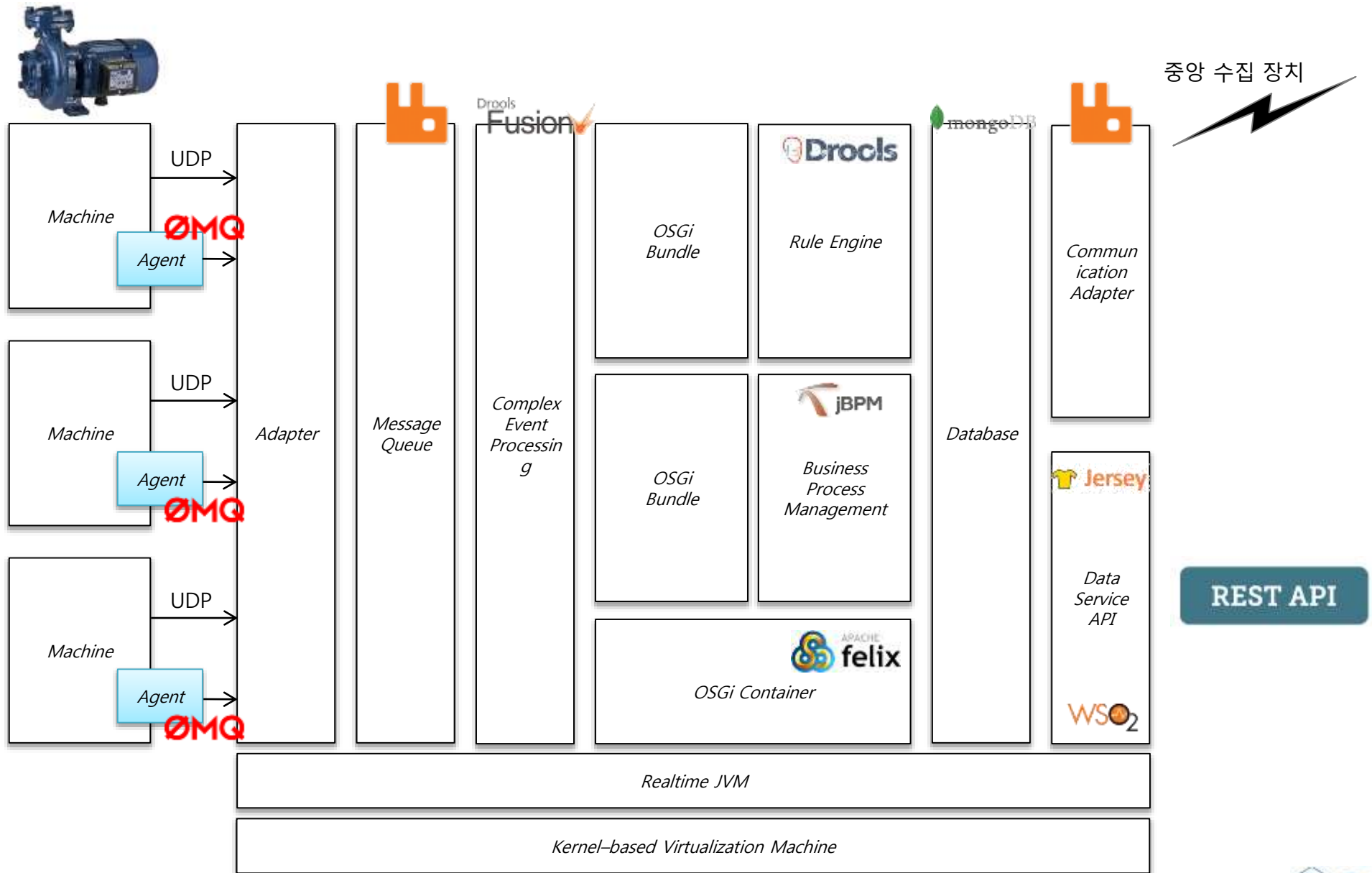
문제: 배치 처리로 인한 분석 시간 지연

해결: ElasticSearch(또는 Splunk)를 통해 Data Discovery 기능 제공

데이터 분석

문제: 고가의 분석 시스템
해결: Pentaho ETL, OLAP 및 PostgreSQL DB를 통한 데이터 분석 및 마트 DB 생성

M2M IoT 데이터 수집



Lesson Learned

데이터 수집 Agent/Adapter

문제: 기계 장비에 부하를 주지 않는
light 하면서도 빠른 통신 필요

해결: ZeroMQ를 통한 가볍고 빠른 통신.
장비에 따라 TCP, UDP 선택

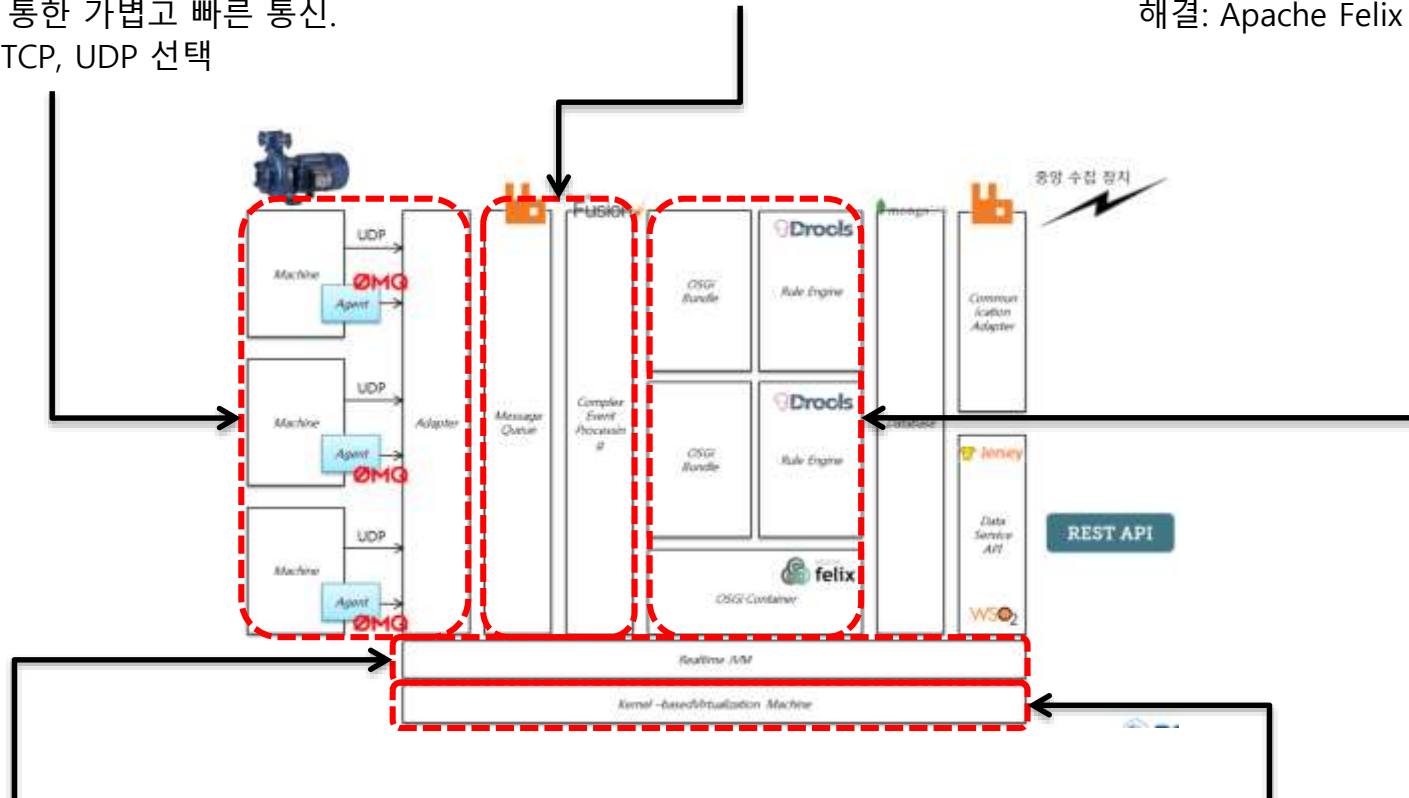
Complex Event Processing

문제: 실시간 데이터 패턴 감지의 어려움
해결: Drools Fusion을 CEP 기능 구현

OSGi를 이용한 다양한 모듈 운용

문제: 모듈에 대한 의존 관계 및 동적인
업데이트의 어려움

해결: Apache Felix 기반의 OSGi 이용



Low Latency 처리

문제: 일부 기계 제어의 경우 응답
시간 보장이 필요

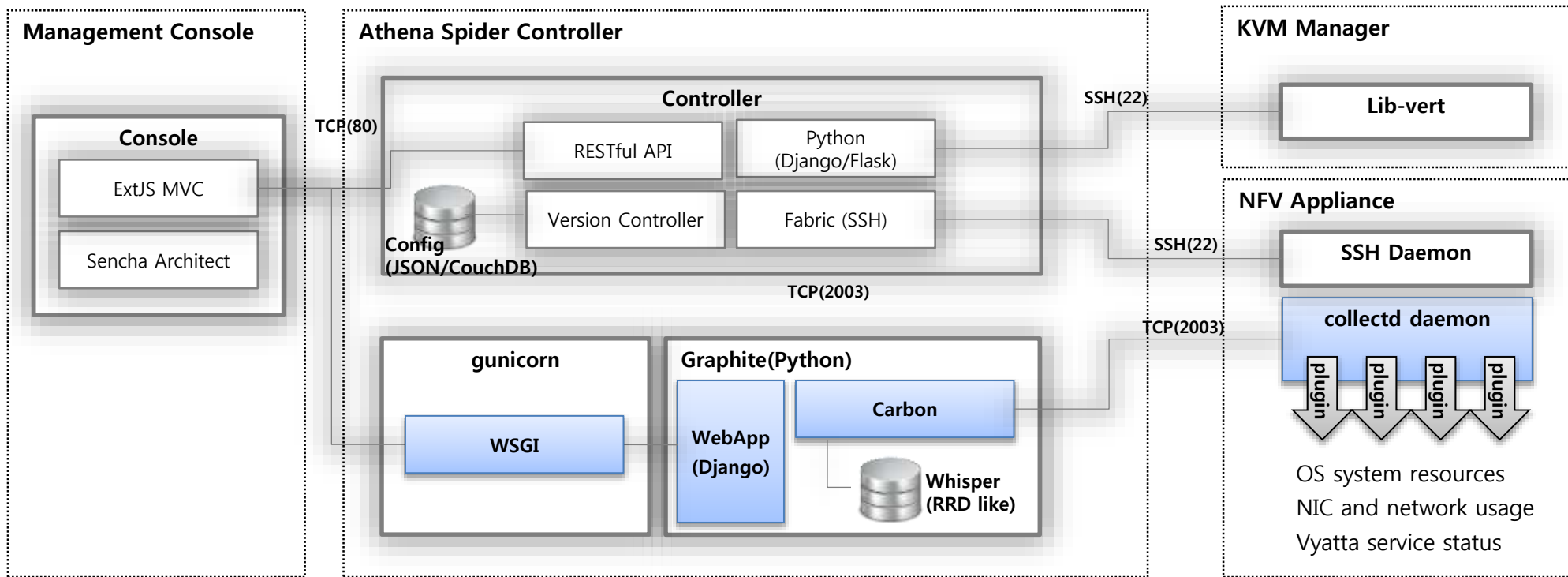
해결: Realtime JVM을 통해 응답
시간 보장

가상화를 통한 Provisioning

문제: 복잡한 컴포넌트들 설치/구성
때문에 신규 시스템 구축 어려움

해결: Linux KVM 가상화를 이용하여
VM 단위 Provisioning

VM 기반 NFV 제어 솔루션



Component	Function	Description
Management Console	<ul style="list-style-type: none"> 사용자를 위한 관리 콘솔 Whisper(RRD like)에 저장된 데이터를 REST API로 받아 ExtJS로 렌더링 하거나 Graphite가 직접 렌더링한 결과를 이미지로 화면에 포함 	ExtJS, Sencha
Athena Spider Controller (API Server)	<ul style="list-style-type: none"> Configuration(JSON 포맷으로 저장) 관리 Lib-vert를 통해 KVM Manager을, SSH를 통해 NFV Appliance 제어 collectd에서 수집한 데이터(Whisper)를 JSON으로 변환하여 REST로 서비스로 제공 	Python Django
Graphite - collectd	<ul style="list-style-type: none"> collectd: 시스템 리소스, NIC 상태, Vyatta 정보를 주기적으로 읽어 Controller로 수집 Graphite: collectd가 보낸 데이터를 Carbon을 통해 Whisper(RRD)에 저장하고 결과를 그래프로 렌더링 	gunicorn을 통해 WSGI 서비스

Lesson Learned

웹 기반 인터페이스

문제: 웹 기반 UI 구현

해결: ExtJS 사용

VM 제어 기능 구현 서버

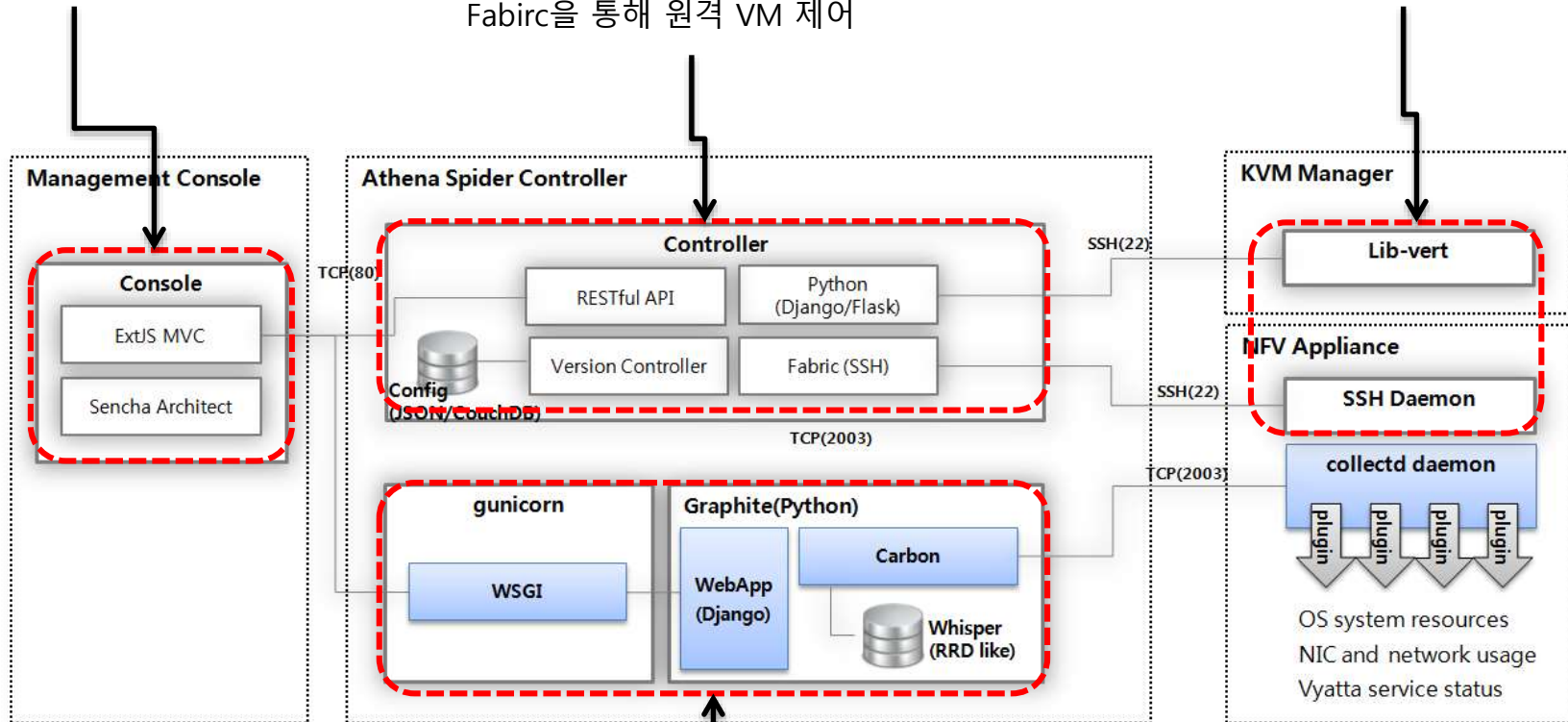
문제: 다양한 VM 제어 기능 구현

해결: Python/Flask 기반 REST API 서버,
Fabric을 통해 원격 VM 제어

VM Provisioning 및 관리

문제: VM 제어 방안 필요

해결: lib-virt 기반 Hypervisor 제어



Graphite/Django 기반 모니터링 UI

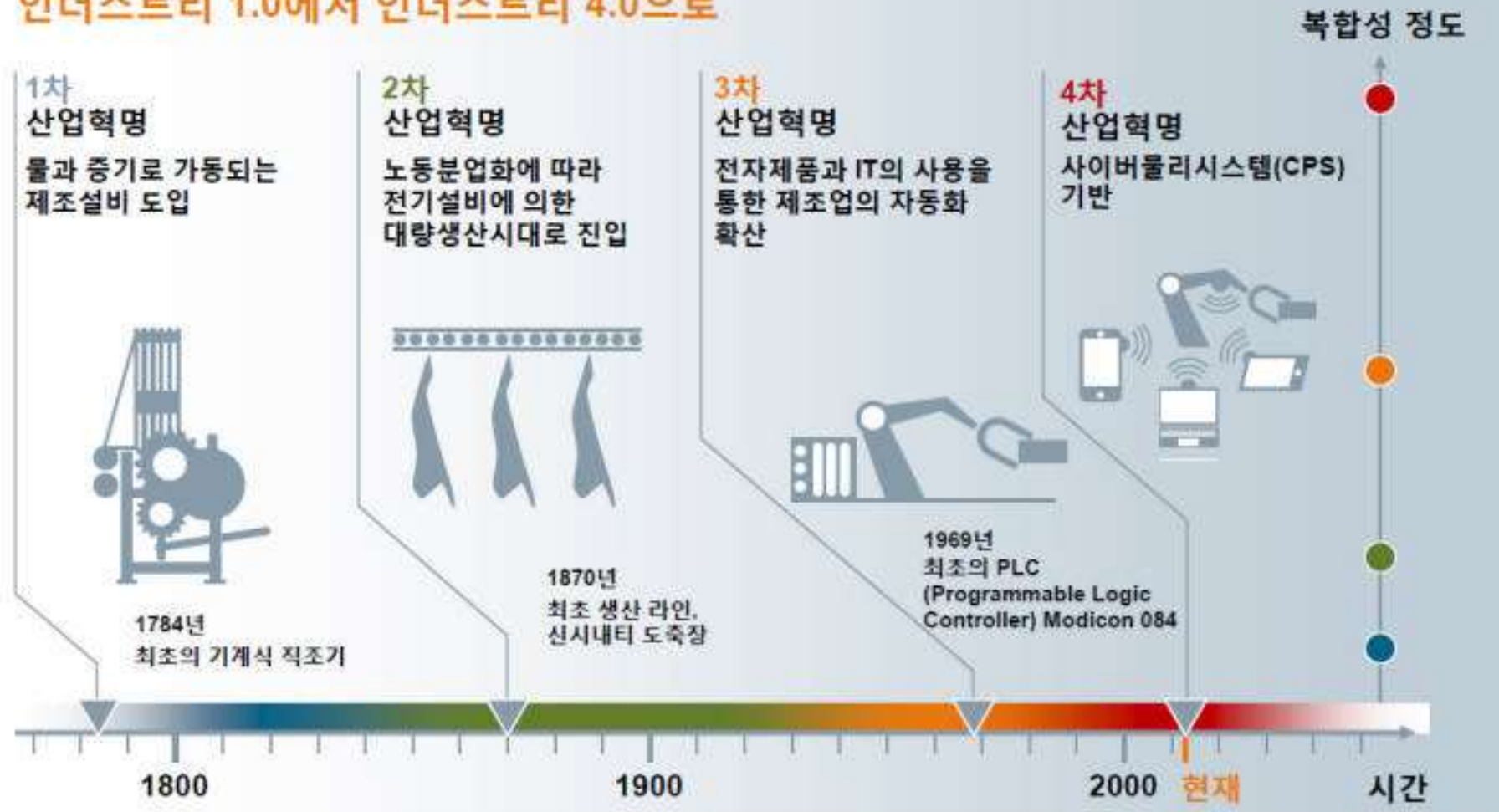
문제: 다양한 모니터링 정보의 시각화

해결: Graphite/Django를 이용하여 그래프
렌더링



산업과 오픈소스의 결합

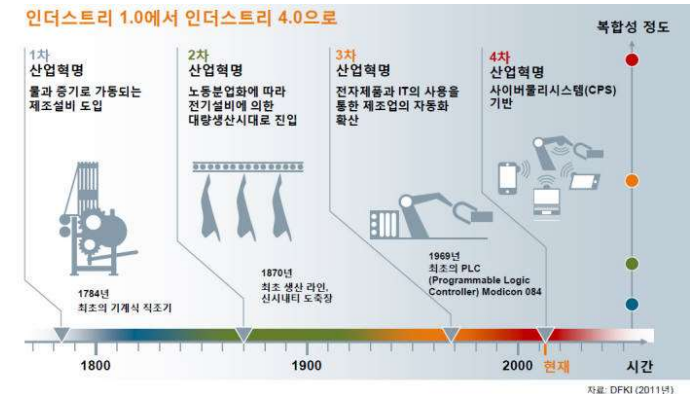
인더스트리 1.0에서 인더스트리 4.0으로



자료: DFKI (2011년)

인더스트리 4.0

- 2008년 금융 위기 이후, 선진국의 제조업 부활 정책 총공세 및 제조업의 중요성 재조명
- ICT 기반 제 4차 산업혁명
- 사물인터넷·빅데이터·클라우드컴퓨팅·스마트 로봇 등 기반 기술의 동시다발적 발전
- 제조 생태계와 초 연결 사회 간의 실시간 연계·소통이 가능
- 제조 강국의 세대교체 가속화, 상품 수출과 기술 서비스 접목 활성화

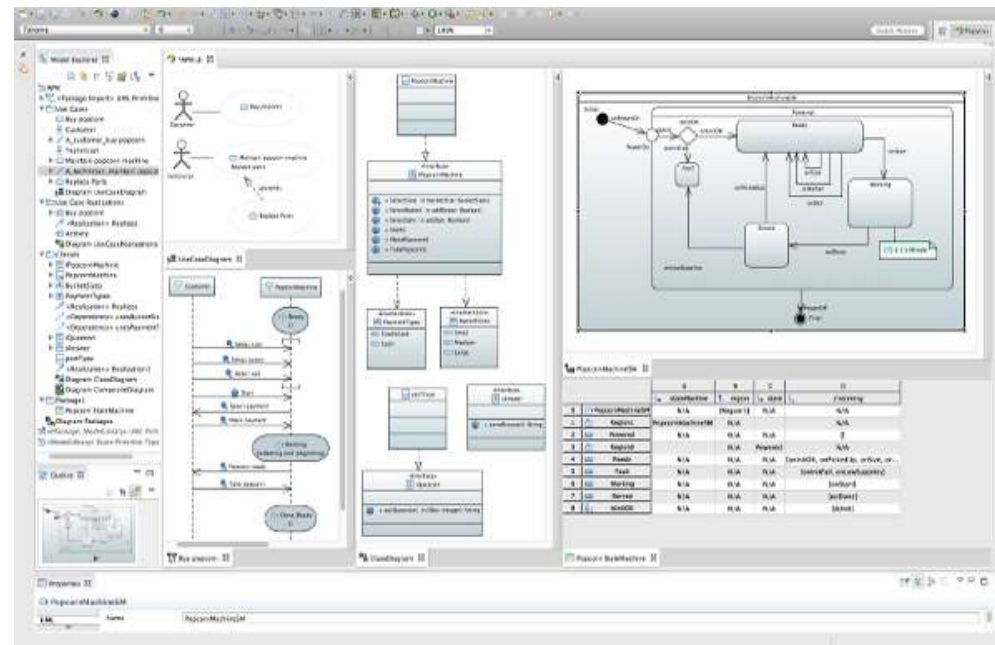
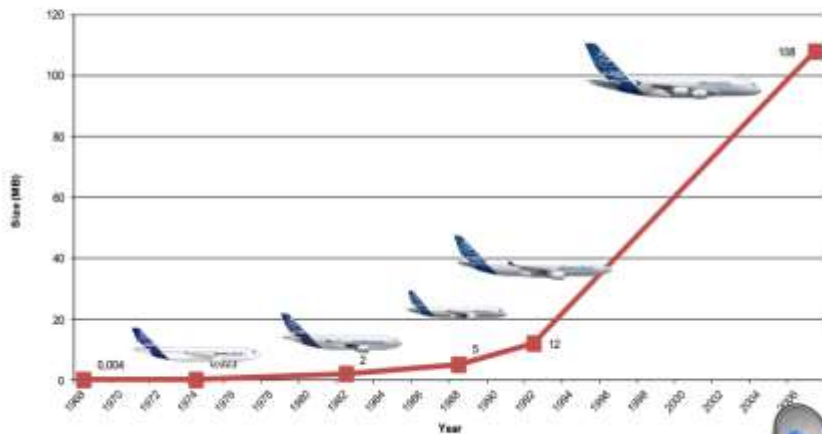


오픈소스 기반 ICT 기술을 통한 산업의 발전

오픈소스의 산업계 적용 예 – Polarsys

- Airbus와 Eclipse Foundation이 주도가 되어 안전과 관련된 산업 기기, 임베디드 소프트웨어 시스템을 오픈소스 기반으로 개발하기 위한 프로젝트
- 항공, 국방, 운송, 통신, 에너지, 의료 등에서 필요로 하는 개발 도구
- 국제적인 규제에 근거하여 요구사항 수집부터 개발, 테스트까지 제품에 대한 전체적인 lifecycle을 관리할 수 있는 도구
- 30~70년 유지보수가 가능해야 함 ➔ 개방형 오픈소스 소프트웨어의 필요성

Airbus Aircraft Embedded Software



신문 기사에 언급된 우리 제조업



한국 '세계1위 기술' 한 개뿐

단위:개, 142개 기술 중 1위 품목
자료:한국산업기술평가관리원



중국이 한국보다 앞선 전략 기술

단위:년, 기술 격차

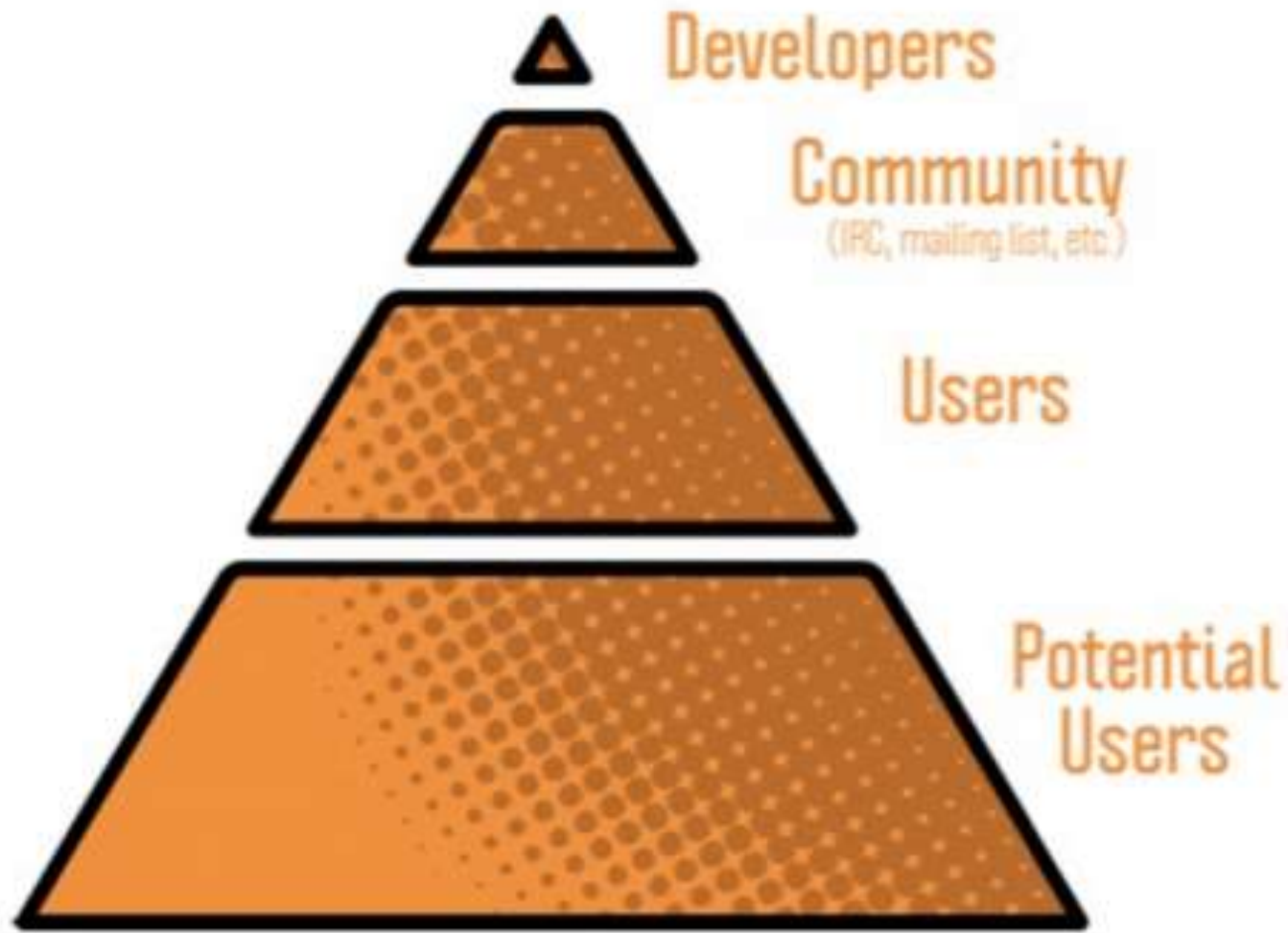
자료:한국경제연구원



국내에는 원천 기술보다는 활용/생산/설비 기술 → 서비스로 발전 필요성



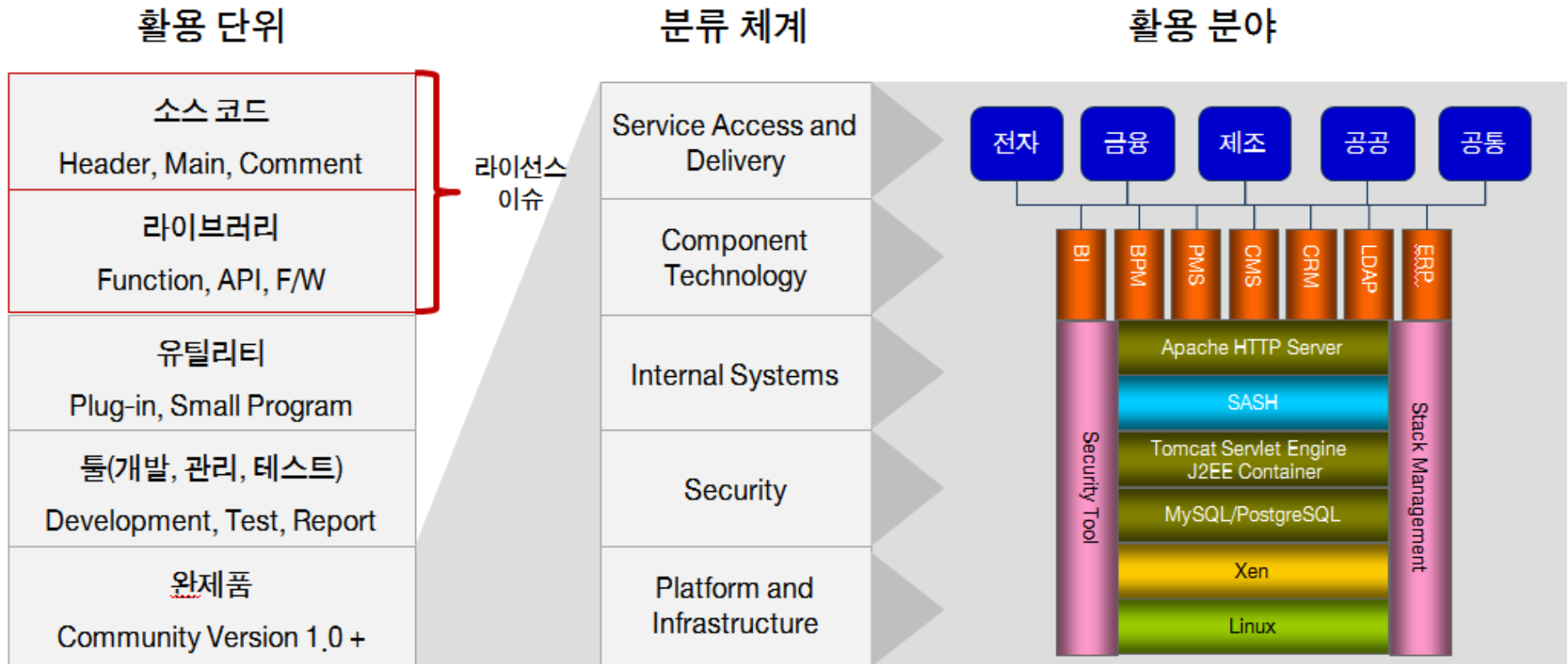
접근 방법 및 프로세스



오픈소스 적용 모델

- OSS의 강자들은 활용 단위에 따라 다양한 방법으로 사업 경쟁력을 강화시킴
- 상용 SW와 OSS의 분류 체계는 유사하며, 완제품 수준의 대체와 연계가 핵심임
- 라이선스 이슈는 보통 소스 코드와 라이브러리 수준에서 결합 형태에 따라 달라짐

※ AGPL은 완제품 수준에서 이슈가 발생함



오픈소스 프로젝트 찾기

- 오픈소스 소스 저장소(Repository)

- GitHub
- BitBucket
- Google Code
- Code Project
- SourceForge
- Naver nForge

- GitHub 상위 랭킹 프로젝트

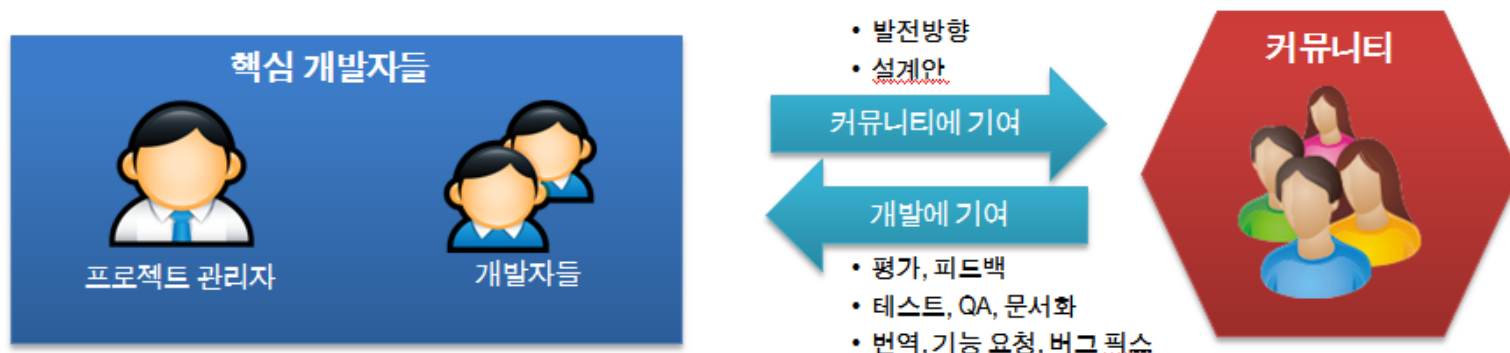
- Most Starts: <https://github.com/search?q=stars:%3E1&s=stars&type=Repositories>
- Most Forks: <https://github.com/search?o=desc&q=stars%3A%3E1&s=forks&type=Repositories>

- 오픈소스 프로젝트 찾기

- <http://www.findbestopensource.com>
- <https://www.openhub.net/>

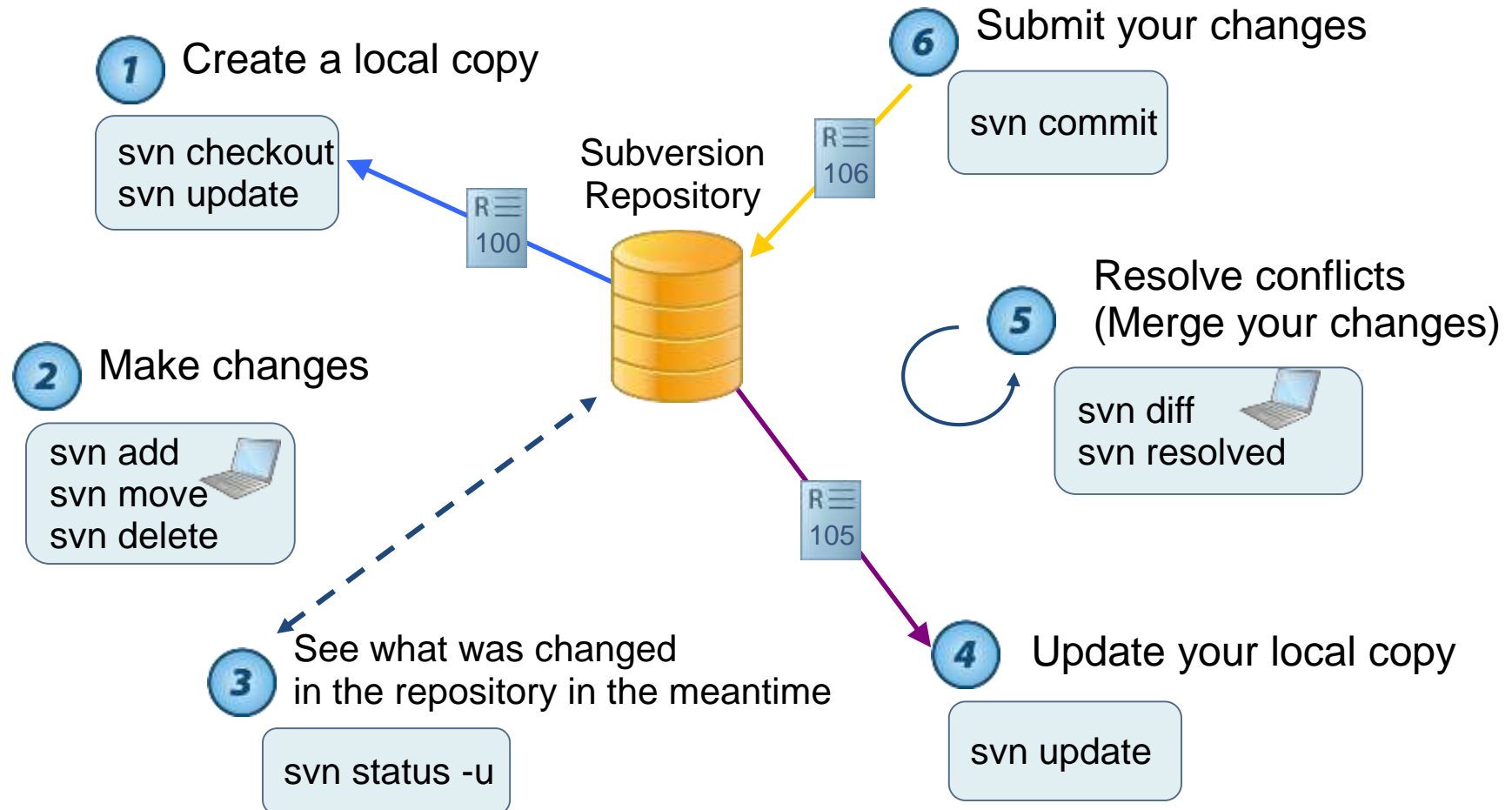
공동 개발 방식으로 소프트웨어 개발

- 하나의 프로젝트 혹은 소프트웨어를 **전 세계에 흩어져 있는 다양한 사람들과 조직이 어우러져서 개발을 진행**하는 방식
- 공동관심사를 갖는 사람들이 함께 특정 소프트웨어나 프로젝트를 개발하는 **관계로 자사의 개발과 투자 비용을 절감하고, 기업 외부의 인력과 기술을 내재화할 수 있는 특성이 있음**
 - 외부 인력과 기술을 활용하여 개발 비용을 절감하기 위해서는, 해당 기술을 이해하는 많은 외부 인력이 존재하고 이들이 오픈 소스 소프트웨어를 통해 자신의 기술과 능력을 공유할 정도의 환경과 규모가 필요
- 공동 개발 방식은 소위 **네트워크 효과(Network Effect)**에 의해 크게 영향을 받는 개발 방식으로 특정 **임계점(Tipping Point)** 이상의 개발자가 소프트웨어 개발에 참여해야만 성공적으로 소프트웨어가 유지, 관리, 개발되는 특성이 있음



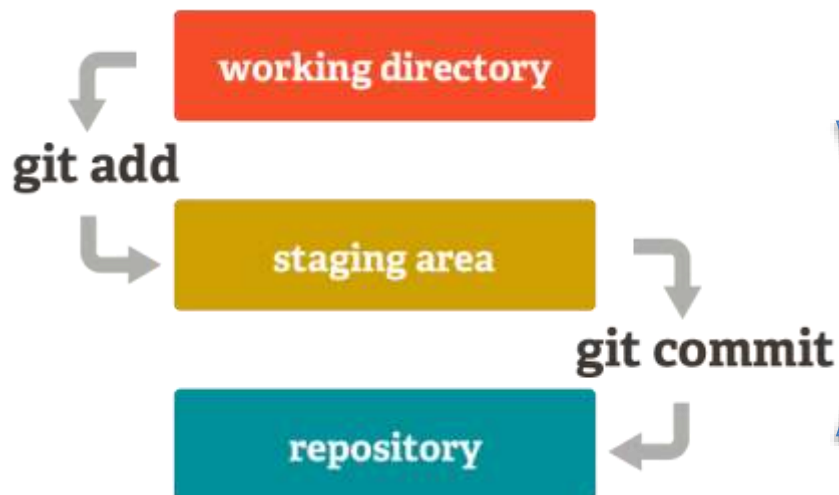
Subversion 작업유형

- 중앙 리포지토리 내에서 로컬 복사 후 변경한 사본을 커밋하는 방식으로 진행



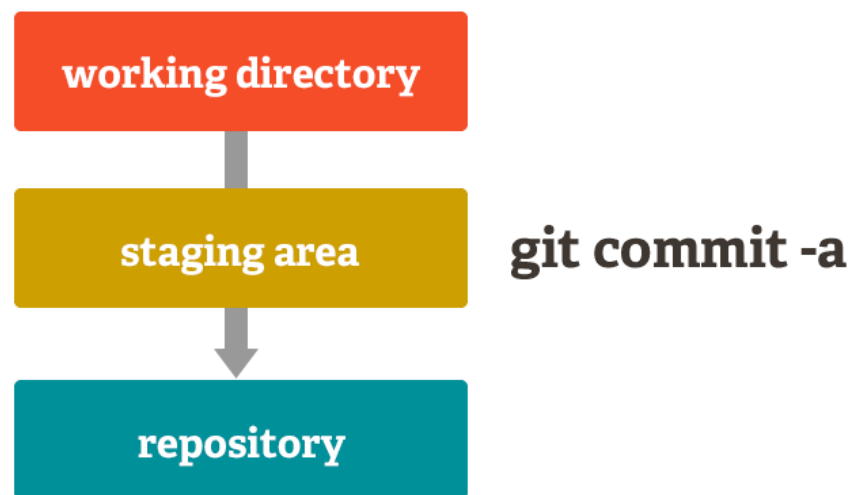
Git 작업 유형

- 기존의 버전 관리 시스템과는 다른 로컬 영역의 저장소의 2단계를 가짐

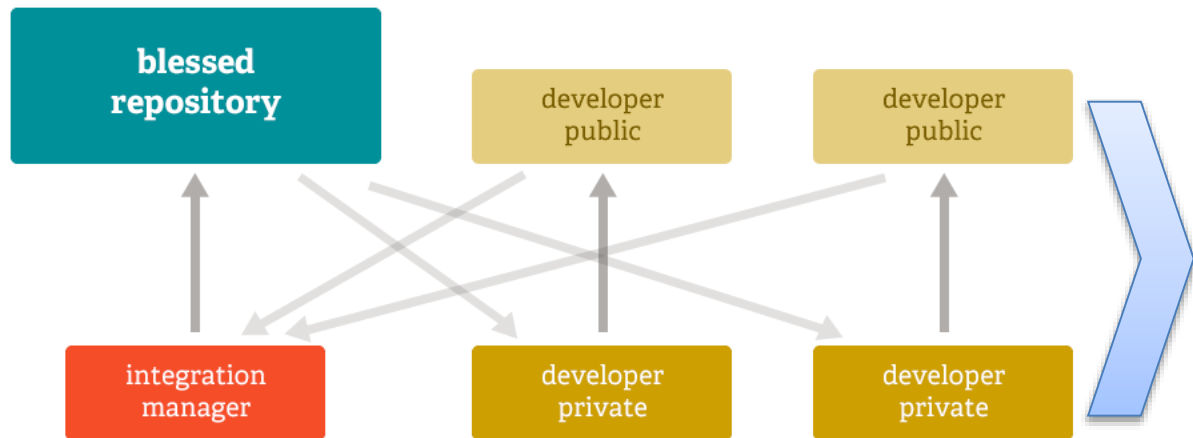


- 명령어들을 로컬에서 실시하고 나중에 서버에 올리는 방식을 사용
- 기존의 svn에서 commit이 1. 소스를 수정하고, 2. 저장소로 업로드 하는 2 단계를 각각의 명령어로 분리하여 중간에 로컬에서 일하다가 저장소에 적용하기 전의 잠재적인 한 번의 단계를 거침

- 한 번의 명령으로 바로 적용되는 SVN과는 달리 2번의 명령을 실행해야 마침내 서버에 있는 저장소로 업로드
- "git commit -a" 의 명령을 통해서 다른 버전관리 프로그램들과 같이 이용하는 것도 가능



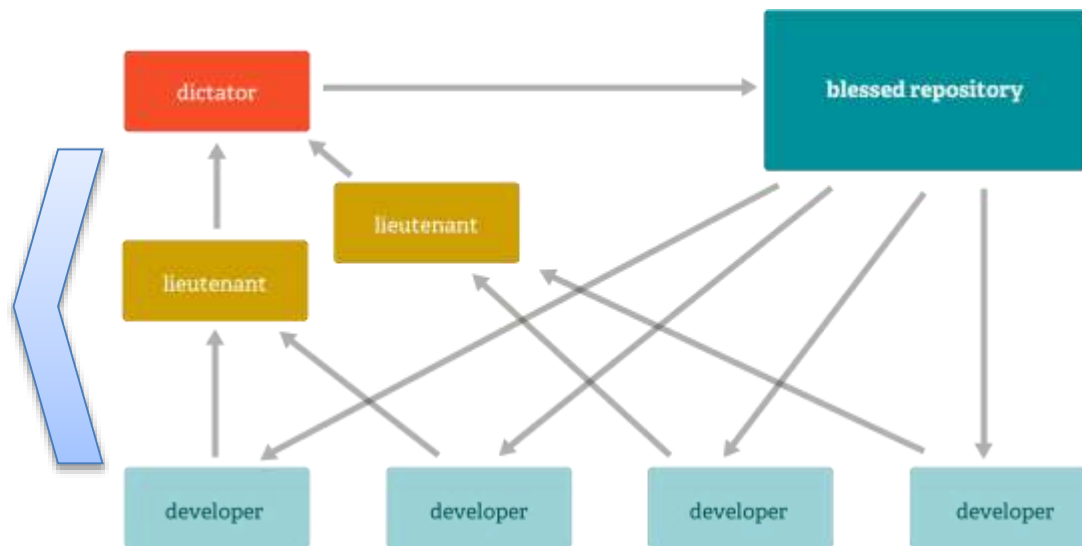
Git 기반 분산형 개발 아키텍처



- 각 팀원의 역할을 분담해서 한 명은 통합을 전담으로 수행
- 다른 개발자는 각자 부분 개발에 집중을 하는 식으로 역할 분담

<통합 관리 워크플로우 모델>

- 팀이 커질수록 복수의 repository를 쉽게 구성 가능
- 하위 팀들을 묶어주는 통합 관리자, 그리고 전체를 통합하는 관리자로 더 세분화해서 구성하는 것이 가능





오픈소스 고려 사항

오픈소스 라이선스 – OSI 인증 70여 개 라이선스

Academic Free License 3.0 (AFL-3.0)
Aferro GNU Public License: See *"GNU Aferro General Public License 3.0 (AGPL-3.0)"*
Adaptive Public License (APL-1.0)
Apache License 2.0 (Apache-2.0)
Apple Public Source License (APSL-2.0)
Artistic license 2.0 (Artistic-2.0)
Attribution Assurance Licenses (AAL)
BSD 3-Clause "New" or "Revised" License (BSD-3-Clause)
BSD 2-Clause "Simplified" or "FreeBSD" License (BSD-2-Clause)
Boost Software License (BSL-1.0)
Computer Associates Trusted Open Source License 1.1 (CATOSL-1.1)
Common Development and Distribution License 1.0 (CDDL-1.0)
Common Public Attribution License 1.0 (CPAL-1.0)
CUA Office Public License Version 1.0 (CUA-OPL-1.0)
EU DataGrid Software License (EUDatagrid)
Eclipse Public License 1.0 (EPL-1.0)
Educational Community License, Version 2.0 (ECL-2.0)
Eiffel Forum License V2.0 (EFL-2.0)
Entessa Public License (Entessa)
European Union Public License, Version 1.1 (EUPL-1.1) (*links to every language's version on their site*)
Fair License (FAIR)
Framework License (Framework-1.0)
GNU Affero General Public License v3 (AGPL-3.0)
GNU General Public License version 2.0 (GPL-2.0)
GNU General Public License version 3.0 (GPL-3.0)
GNU Library or "Lesser" General Public License version 2.1 (LGPL-2.1)
GNU Library or "Lesser" General Public License version 3.0 (LGPL-3.0)
Historical Permission Notice and Disclaimer (HPND)
IBM Public License 1.0 (IPL-1.0)
IPA Font License (IPA)
ISC License (ISC)
LaTeX Project Public License 1.3c (LPPL-1.3c)
Lucent Public License Version 1.02 (LPL-1.02)
MirOS Licence (MirOS)

Microsoft Public License (Ms-PL)
Microsoft Reciprocal License (Ms-RL)
MIT license (MIT)
Motosoto License (Motosoto)
Mozilla Public License 2.0 (MPL-2.0)
Multics License (Multics)
NASA Open Source Agreement 1.3 (NASA 1.3)
NTP License (NTP)
Naumen Public License (Naumen)
Nethack General Public License (NGPL)
Nokia Open Source License (Nokia)
Non-Profit Open Software License 3.0 (NPOSL-3.0)
OCLC Research Public License 2.0 (OCLC-2.0)
Open Font License 1.1 (OFL 1.1)
Open Group Test Suite License (OGTSL)
Open Software License 3.0 (OSL-3.0)
PHP License 3.0 (PHP-3.0)
The PostgreSQL License (PostgreSQL)
Python License (Python-2.0) (overall Python license)
CNRI Python license (CNRI-Python) (CNRI portion of Python License)
Q Public License (QPL-1.0)
RealNetworks Public Source License V1.0 (RPSL-1.0)
Reciprocal Public License 1.5 (RPL-1.5)
Ricoh Source Code Public License (RSCPL)
Simple Public License 2.0 (SimPL-2.0)
Sleepycat License (Sleepycat)
Sun Public License 1.0 (SPL-1.0)
Sybase Open Watcom Public License 1.0 (Watcom-1.0)
University of Illinois/NCSA Open Source License (NCSA)
Vovida Software License v. 1.0 (VSL-1.0)
W3C License (W3C)
wxWindows Library License (WXwindows)
X.Net License (Xnet)
Zope Public License 2.0 (ZPL-2.0)
zlib/libpng license (Zlib)

오픈소스 라이선스 – 특징(양면성)

자유

복제, 설치, 운영,
수정, 배포의 자유

- 기존 소스 코드 및 라이브러리의 재사용으로 **생산성 향상**
- 성숙한 OSS 커뮤니티의 프로젝트 적용으로 **품질 보장**

Copyleft

공개되어 있는
소스 코드

- 공개 되어 있는 소스 코드를 통해 알고리즘과 로직을 분석하고 응용할 수 있어 설계자, 개발자, 테스터의 **소프트웨어 기술력 증강**

무료

내부 사용

외부 판매

- **TCO(Total Cost Ownership) 절감**
- 기술지원 및 유지보수 서비스를 기반으로 **사업 가능**

반환 의무

특정 OSS
라이선스의 소스
코드 공개 의무

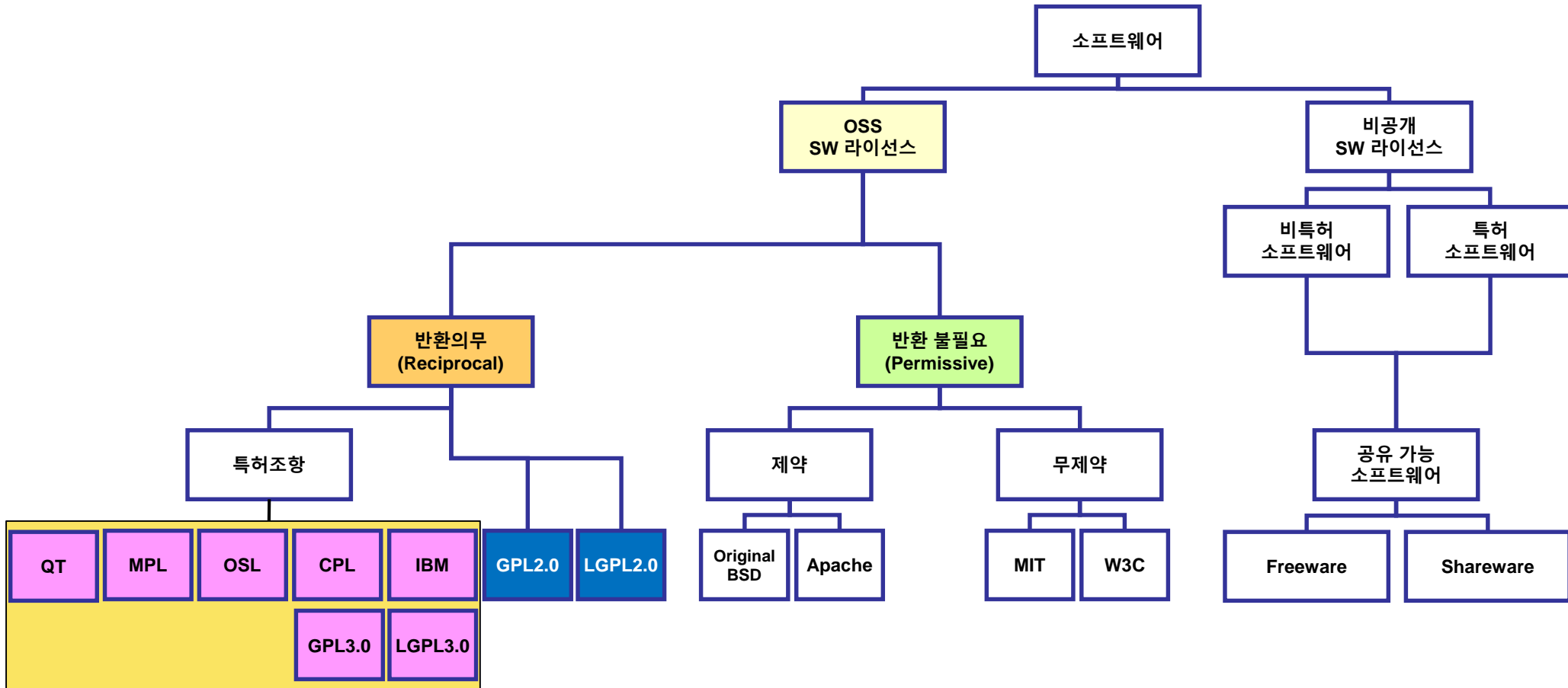
- GPL, LGPL, MPL, CPL, IBM, EPL, OSL, Qt 등이 해당됨
- 연결된 소프트웨어의 소스 코드 **공개의무**가 발생
- 비공개 SW가 **OSS로 변경**될 수 있음
- 특허, 영업비밀, 핵심기술 등의 **외부 유출** 가능성 존재

저작권

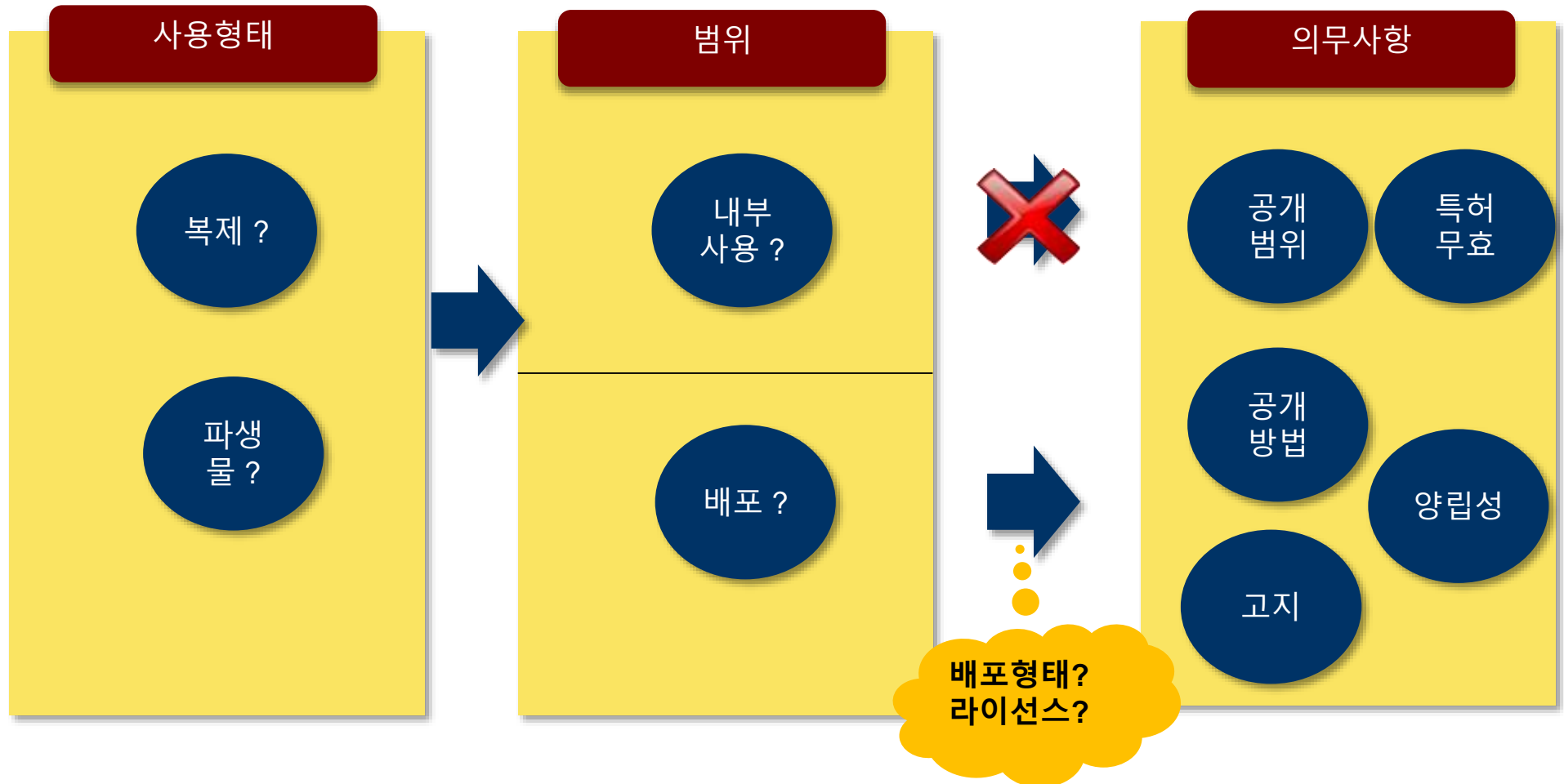
저작권법에 따른
법적 권리 보장

- OSS 라이선스 미 준수 시 저작자와 **분쟁 가능**
- 협력업체가 저작권 위반 시 협력업체가 책임을 지더라도 최종적으로 고객이 동의하지 않는 소스 코드 공개 등은 **공급사의 부담**이 됨

오픈소스 라이선스 - 분류



오픈소스 라이선스 – 주요용어 및 개념



오픈소스 라이선스 – 사용빈도 Top 20

Rank	License	%
1.	GNU General Public License (GPL) 2.0	26%
2.	MIT License	18%
3.	Apache License 2.0	15%
4.	GNU General Public License (GPL) 3.0	11%
5.	BSD License 2.0 (3-clause, New or Revised) License	7%
6.	Artistic License (Perl)	5%
7.	GNU Lesser General Public License (LGPL) 2.1	5%
8.	GNU Lesser General Public License (LGPL) 3.0	2%
9.	http://www.opensource.org/licenses/ms-pl	2%
10.	Eclipse Public License (EPL)	2%
11.	Code Project Open License 1.02	1%
12.	Mozilla Public License (MPL) 1.1	< 1%
13.	Simplified BSD License (BSD)	< 1%
14.	Common Development and Distribution License (CDDL)	< 1%
15.	Microsoft Reciprocal License	< 1%
16.	GNU Affero General Public License v3 or later	< 1%
17.	Sun GPL With Classpath Exception v2.0	< 1%
18.	CDDL-1.1	< 1%
19.	zlib/libpng License	< 1%
20.	Common Public License (CPL)	< 1%

[출처 : Black Duck Software, Inc 2014년]

오픈소스 라이선스 및 의무사항

- 주요 OSS 라이선스인 GPL, LGPL, CPL, MPL 등은 OSS 코드 뿐 아니라 사용자 코드 공개의무 발생
- OSS를 활용하면서 공개하기 어려운 코드를 보호하기 위해서는 적절한 라이선스 관리가 필요함

라이선스	주요 의무사항	공개 범위 및 특징
GPL (General Public License)	<ul style="list-style-type: none">자유로운 사용, 복제, 배포 및 수정저작권 표시, 보증책임이 없다는 표시, GPL 명시소프트웨어 수정 및 링크 시 모든 코드를 GPL에 의해 공개	<ul style="list-style-type: none">GPL 코드와 연결된 모든 코드
LGPL (Lesser General Public License)	<ul style="list-style-type: none">자유로운 사용, 복제, 배포 및 수정저작권 표시, 보증책임이 없다는 표시, LGPL 명시소프트웨어 수정 및 링크 시 모든 코드를 LGPL에 의해 공개 (단, 라이브러리 링크 시 공개하지 않음)	<ul style="list-style-type: none">LGPL 코드와 연결된 모든 코드
MPL (Mozilla Public License)	<ul style="list-style-type: none">자유로운 사용, 복제, 배포 및 수정저작권 표시, 보증책임이 없다는 표시, MPL 명시특허 보복조항(특허 SW사용시 특허권을 주장할 수 없음)소프트웨어 수정 및 링크 시 해당 파일을 공개	<ul style="list-style-type: none">MPL 코드와 연결된 해당 파일
EPL (Eclipse Public License)	<ul style="list-style-type: none">자유로운 사용, 복제, 배포 및 수정저작권 표시, 보증책임이 없다는 표시, MPL 명시특허 보복조항(특허 SW사용시 특허권을 주장할 수 없음)소프트웨어 수정 및 링크 시 해당 모듈을 공개	<ul style="list-style-type: none">EPL 코드와 연결된 해당 모듈특허 SW시 특허권리 주장하지 못함

- 라이브러리 교체가 프레임워크 교체에 비해 쉬움
- 운영 체제, 관리/모니터링 컴포넌트 적용 용이
- 미들웨어(웹서버, 앱서버, Queue 등) 교체가 DB 교체에 비해 쉬움
- 데이터 저장, 처리 등 전문/업무 영역은 범용적인 분야에 비해 부족
- 사실상의 표준에 가까운 인지도 높은 오픈 소스 활용
- 유연성과 민첩성 확보를 위한 범용적인 아키텍처의 중요성
- TCO 절감 및 ROI 극대화를 위한 내재화가 필수

오픈소스에 대한 부정적인 의견

- 상용 소프트웨어를 선택하는 것이 유리한 경우
 - 상용의 사용 방법이 훨씬 쉬운 경우
 - 상용이 사실상의 표준인 경우
 - 상용의 지원이 우수하거나 기능이 더 풍부한 경우
 - SaaS 기반 서비스가 필요할 때
 - 하드웨어 호환성이 우수한 경우(예: Window XP HW Driver 지원)
 - 보증/책임보상이 중요한 경우
- 경우에 따른 선택이 중요
 - 기능이나 호환성 같은 명확히 우수한 요건의 경우에는 상용 선택이 유리
 - 지원과 같은 비기능적인 요건의 경우에는 소스가 공개된 오픈소스가 유리할 수도 있음

80%는 오픈소스의 품질이 상용 제품보다 나은 것으로 평가

Source: 2015 Future of Open Source Survey Results

● 오픈소스의 보안

- Linux가 타 플랫폼에 비해서 안정적이라는 보고
- 제로데이어택(아직 밝혀지지 않은 버그를 이용한 공격) 시대라는 관점에서, 공개된 소스를 통해 문제점을 빨리 찾을 수 있는 오픈소스가 보안성 높음(Gartner)
- 소스가 공개되어 있어 투명하고 문제점 발견 시, 소수에 의해 주도되는 프로젝트에 비해 빠르게 해결 가능

72%가 오픈소스가 상용 제품들보다 보안이 우수하다 평가

Source: 2015 Future of Open Source Survey Results



결론

오픈소스 도입에 대한 SWOT 분석

● SWOT 분석 적용을 통해 각 요소 별 항목을 정리

역량	시점	잠재력과 경쟁력(중점과제)	한계와 문제점(극복과제)
	현재환경(내부환경)	강점요소(Strength)	약점요소(Weakness)
	미래환경(외부환경)	기회요소(Opportunity)	위협요소(Threat)

강점요소(Strength)	약점요소(Weakness)
<ul style="list-style-type: none"> 전세계의 다양하고 많은 기업들이 기술 개발에 참여 X86 시스템과 함께 높은 기술 경쟁력 및 시장 점유율 보유 가상화/클라우드 컴퓨팅 기반 기술을 제공하며 시스템 혁신 가능 국내외 기술지원 전문기업과 엔지니어가 많아 지고 있음 	<ul style="list-style-type: none"> 오픈 소스에 대한 이해도, 생태계, 개발 프로세스 등에 대한 인식이 부족 정보 기획/운영 관점의 투자(초기교육, 컨설팅)가 필요한 분야 복잡한 라이선스 모델
기회요소(Opportunity)	위협요소(Threat)
<ul style="list-style-type: none"> 핵심 기술에 대한 기술력(내재화)을 통한 경쟁사 우위 모바일/클라우드/IoT 등 다양한 분야에 대한 빠른 시장 접근 기술 도입 및 운용 비용의 절감 	<ul style="list-style-type: none"> 상용SW 벤더의 라이선스 관리 서비스(LMS)를 통해 법적 이슈를 만들어 지속적으로 Lock-in 시킴 기존의 IT 환경에 익숙한 관리자들의 반발 자체 기술력 부족 시 비용 절감 효과가 낮음

개방형 기술을 통한 사실상의 표준 주도

오픈소스를 활용한 기술과 서비스의 분리

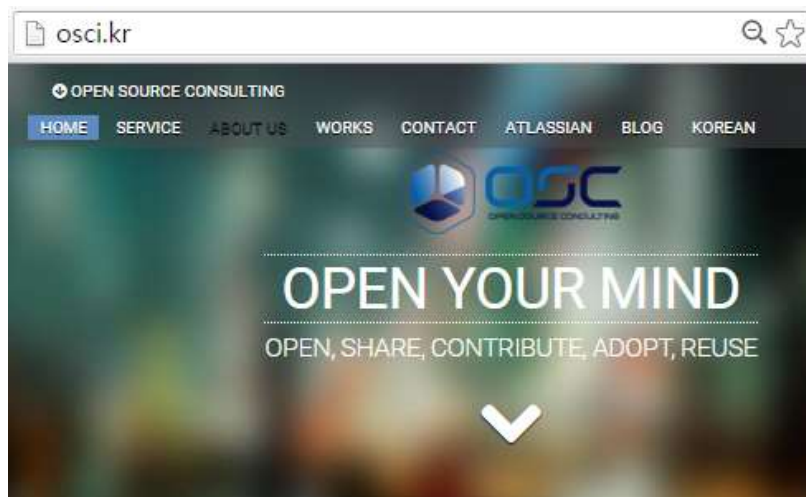
오픈소스는 시대적 흐름



에 대한

이 필요하면

오픈소스 컨설팅
오픈소스 컨설팅으로...



<http://osci.kr>

오픈소스와 열정을 함께 할 인재들을 모십니다

WHO WE ARE - OUR TEAM & ABOUT

About us

Open Source Consulting Inc. is established as a technical and professional consulting service company based out of Seoul. We have various skilled staff specialists across Linux, VMware, Oracle, Open Stack, Ansible, AWS, cloud computing and many Open Source software. One of our major job is to provide software solutions for our clients. We are a team of professionals who are passionate about Open Source software. We are a team of professionals who are passionate about Open Source software. We are a team of professionals who are passionate about Open Source software.

Our vision

Open Source Consulting Inc. is able to boost our own development and management competence through its business. Our vision is to provide innovative solutions which can bridge the gap between companies and customers with guaranteed quality. We provide customers' satisfaction through comprehensive innovation and superior professional service.

Our experience

We have in-depth knowledge in all major areas of open source. One of our 100 members have 10+ years of experience in professional open source services, and our previous customers include more than 20 customers in a wide range of industries. We have completed big data and open source.



LEE JI HUN - OPEN SOURCE CONSULTANT
LEE JI HUN - OPEN SOURCE CONSULTANT
LEE JI HUN - OPEN SOURCE CONSULTANT



LEE JI HUN - OPEN SOURCE CONSULTANT
LEE JI HUN - OPEN SOURCE CONSULTANT
LEE JI HUN - OPEN SOURCE CONSULTANT



LEE JI HUN - OPEN SOURCE CONSULTANT
LEE JI HUN - OPEN SOURCE CONSULTANT
LEE JI HUN - OPEN SOURCE CONSULTANT



LEE JI HUN - OPEN SOURCE CONSULTANT
LEE JI HUN - OPEN SOURCE CONSULTANT
LEE JI HUN - OPEN SOURCE CONSULTANT



LEE JI HUN - OPEN SOURCE CONSULTANT
LEE JI HUN - OPEN SOURCE CONSULTANT
LEE JI HUN - OPEN SOURCE CONSULTANT



LEE JI HUN - OPEN SOURCE CONSULTANT
LEE JI HUN - OPEN SOURCE CONSULTANT
LEE JI HUN - OPEN SOURCE CONSULTANT

recruit@osci.kr

위키피디아

- https://ko.wikipedia.org/wiki/%EC%98%A4%ED%94%88_%EC%86%8C%EC%8A%A4

OSI

- <https://opensource.org/osd>

블랙덕

- 2014 Future of Open Source Survey Results
- 2015 Future of Open Source Survey Results

IDC

- Worldwide Linux Server Operation System Environment by Vendors, 2006-2010, March IDC 2011

IT World

- <http://www.itworld.co.kr/news/97250>

Yahoo Finance

- <http://finance.yahoo.com>

NIPA/MASO

- 공개SW 성공 사례집

Polarsys

- <https://www.polarsys.org/>