



JBoss EAP 7 소개



JBoss EAP

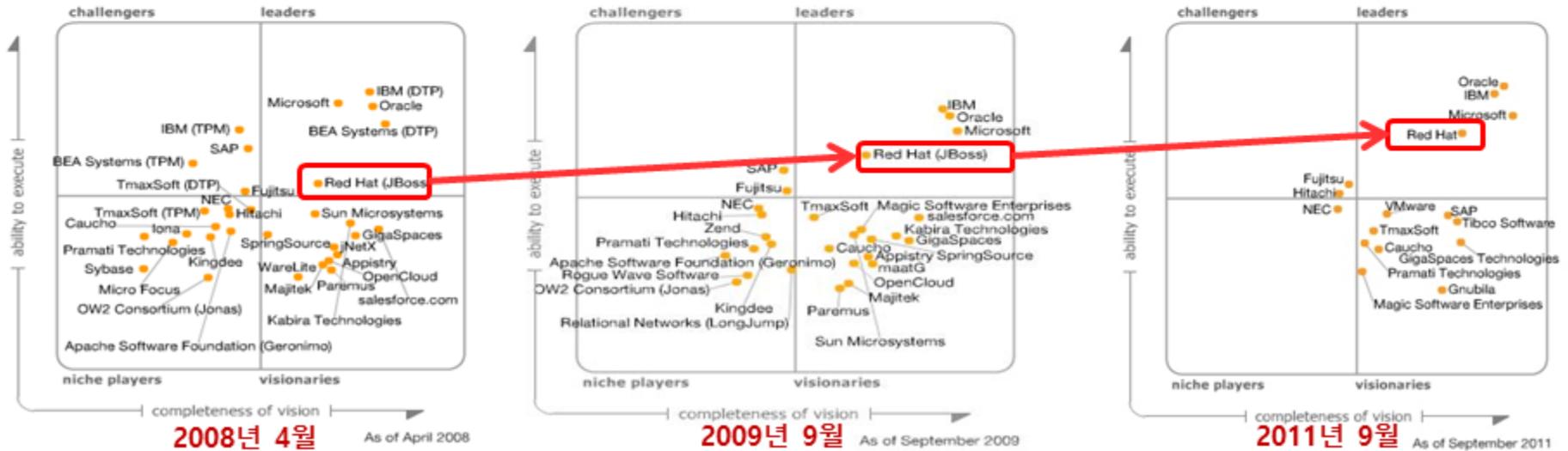
2017년까지 새로운 기업용 자바 어플리케이션의 적어도 70%는 오픈소스 WAS 에서 운영될 것이다.

- 주요 오픈소스 애플리케이션 서버인 Apache와 RedHat 제품이 계속해서 소규모의 Java EE환경을 지배
- 이러한 플랫폼은 후지쯔, 히타치, IBM, 오라클, SAP 및 폐쇄된 소스 솔루션에 비해 낮은 TCO의 인식을 바탕으로 High End 환경으로 확장되고 경쟁
- 많은 클라우드 프로바이더들은 이러한 솔루션을 전략적으로 활용하여 OSS로 전환함으로써 자신들의 SaaS와 PaaS 에 대한 규모의 경제를 실현할 것이다.

Gartner

Predicts 2012: Cloud and In-Memory Drive Innovation in Application Platforms - <http://goo.gl/zpAmU>

JBoss 는 Gartner에서 발표하는 “Magic Quadrant for Application Infrastructure” 에서 확실한 리더로 자리 매김



Source: Gartner (September 2011)



IBM WebSphere AS



JBoss EAP



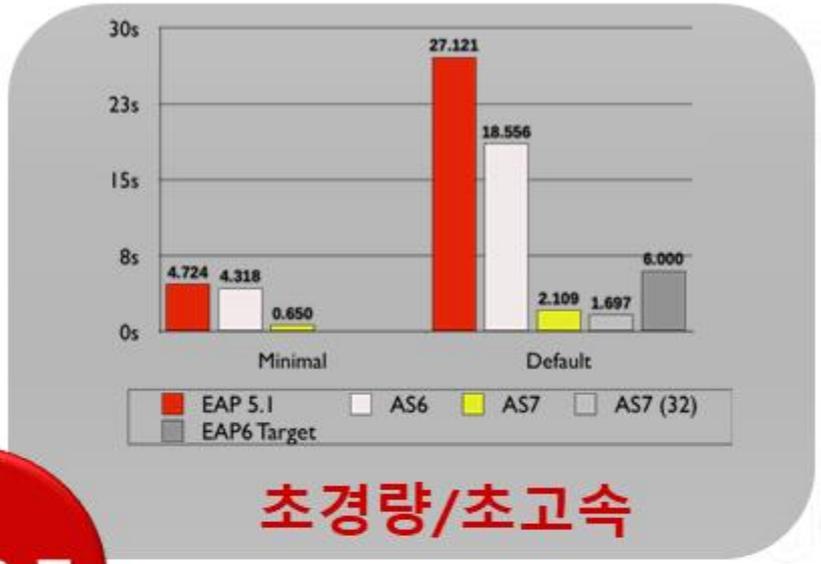
Oracle WebLogic Server

In the war of middleware!



JAVA EE6
Spring
HIBERNATE
GWT
Apache
maven

OSS 프레임워크 지원



EAP 7

풍부한 관리 인터페이스

JBoss OPERATIONS NETWORK
MANAGE
OPENSIFT

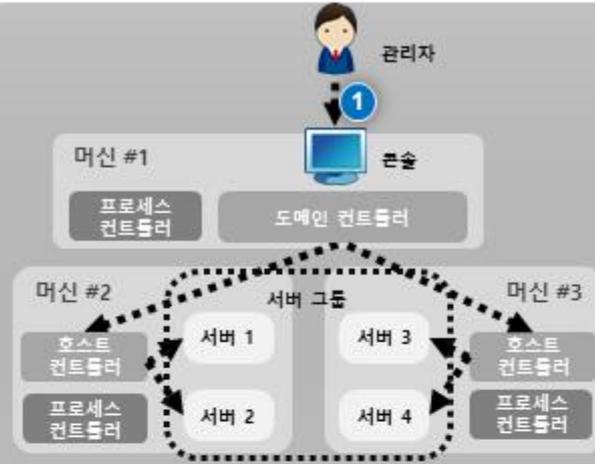
데이터그리드 / 클라우드



JBoss EAP 7 주요 특징



모듈 아키텍처



도메인 모드

EAP 7



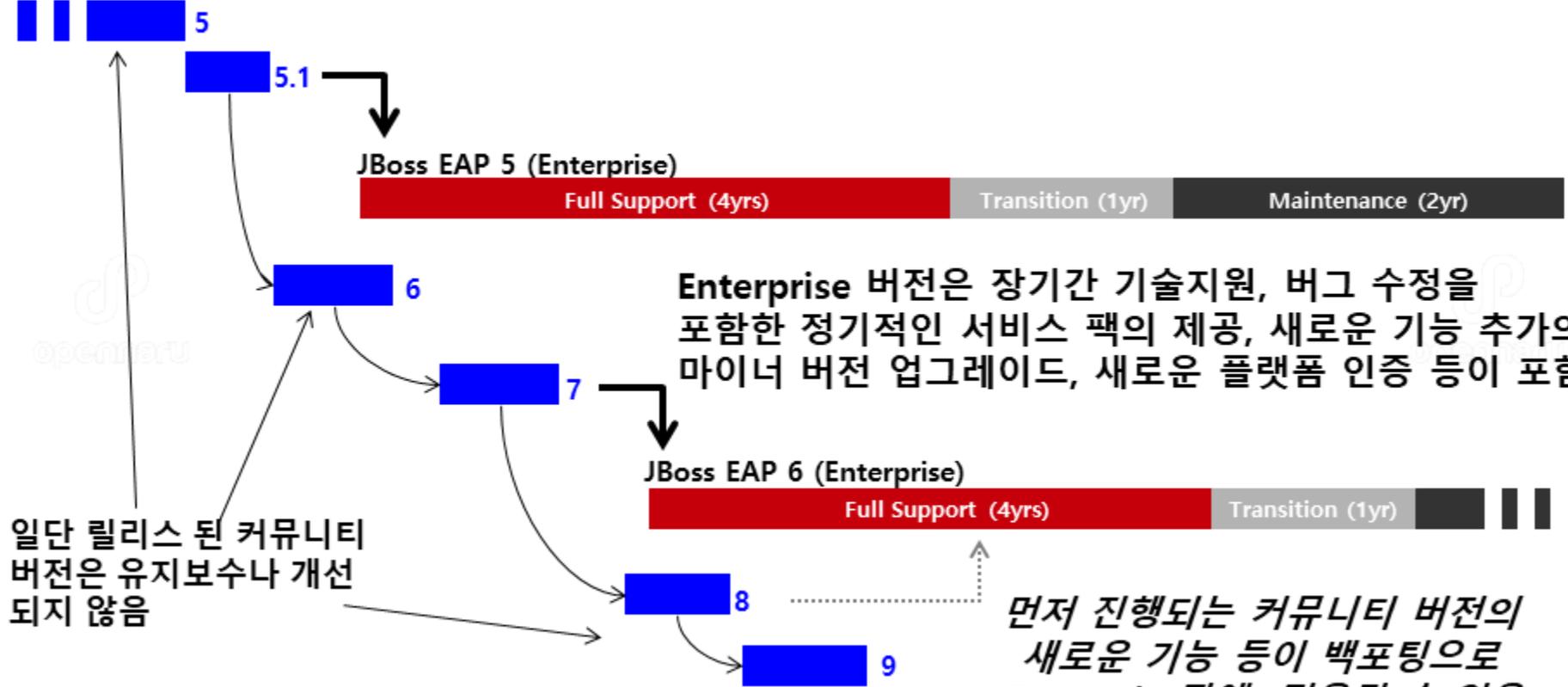
관리 인터페이스



서브시스템

Community과 Enterprise 버전의 관계

JBoss AS (Community)



A close-up photograph of several bright yellow chrysanthemum flowers against a dark background. The flowers are in various stages of bloom, with some showing the intricate details of their petals and centers. The lighting is soft, highlighting the texture and color of the petals.

JBoss Enterprise Middleware

MIDDLEWARE

JBoss
Developer
Studio

Development
Tools

User Interaction

- JBoss Portal

Business Process
Management

- JBoss BRMS
- JBoss BPM Suite

Application
Integration

- JBoss Fuse
- JBoss A-MQ
- JBoss Fuse Service Works

Data Virtualization

- JBoss Data Virtualization

Foundation

- JBoss EAP
- JBoss Web Server
- JBoss Data Grid

- JBoss
Operations
Network

Management
Tools

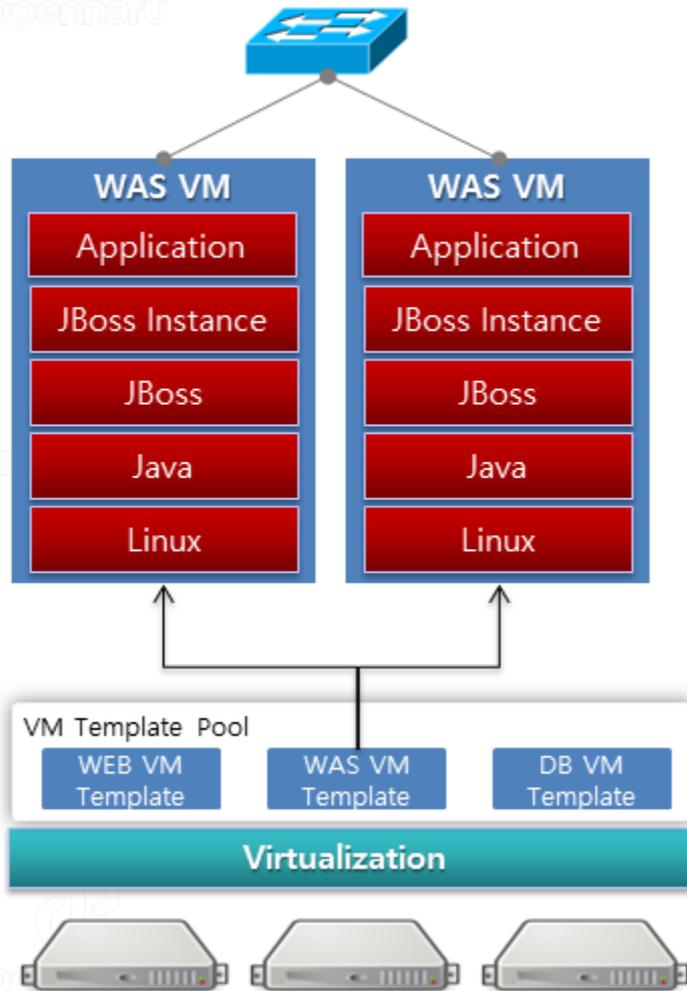
Enterprise Java Yesterday



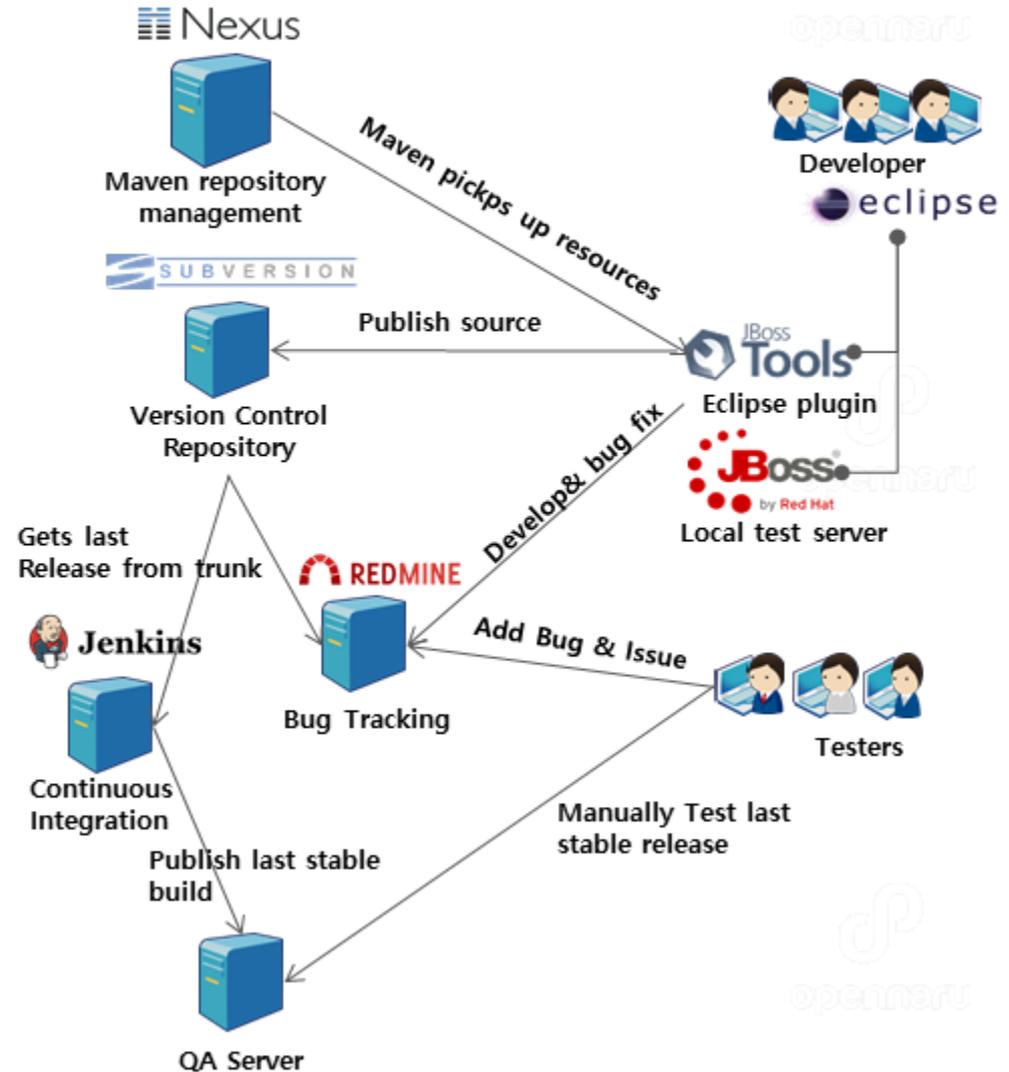
Enterprise Java Today & Tomorrow



운영 / 테스트 환경

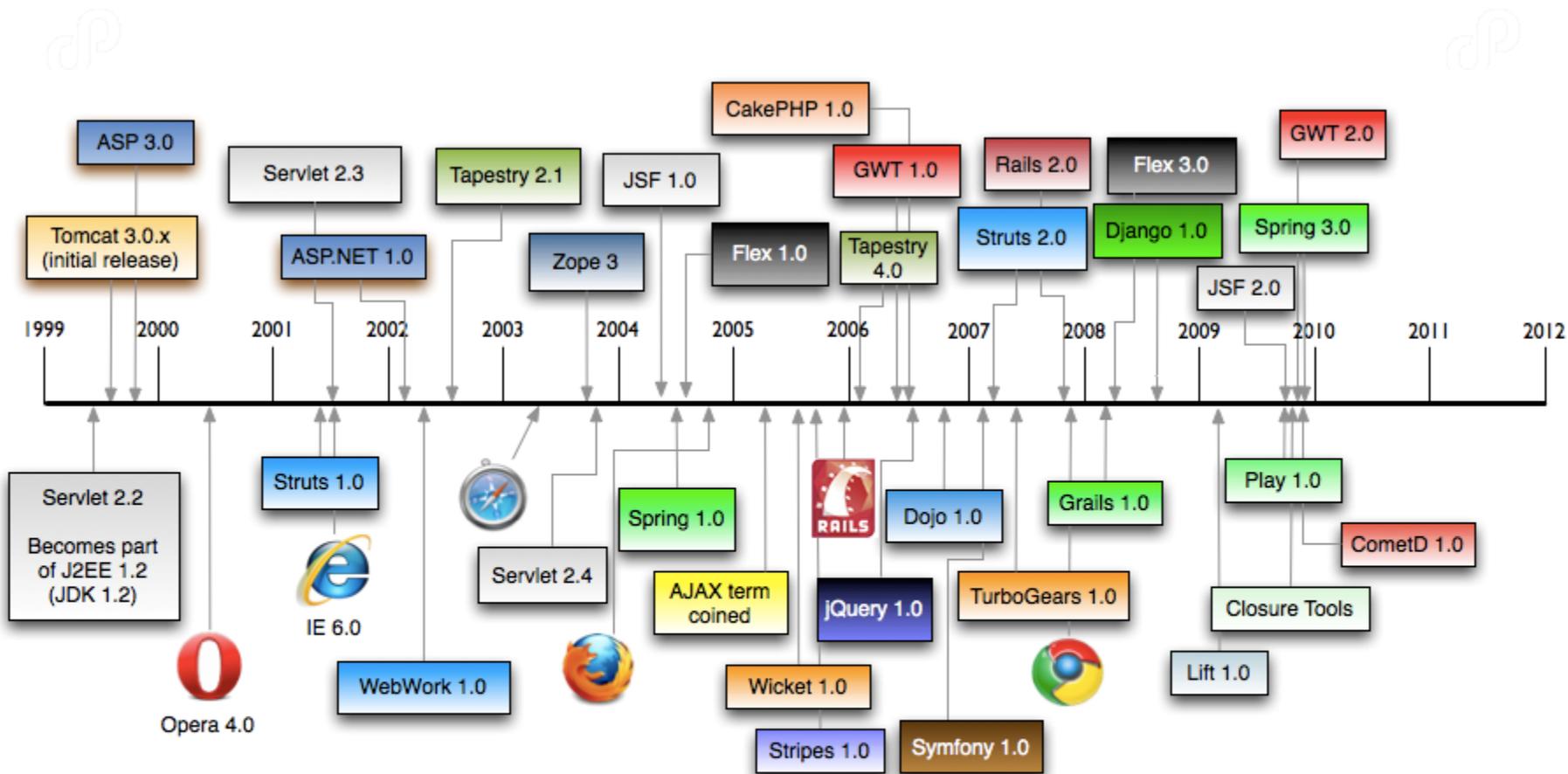


개발 지원 환경



개발 환경

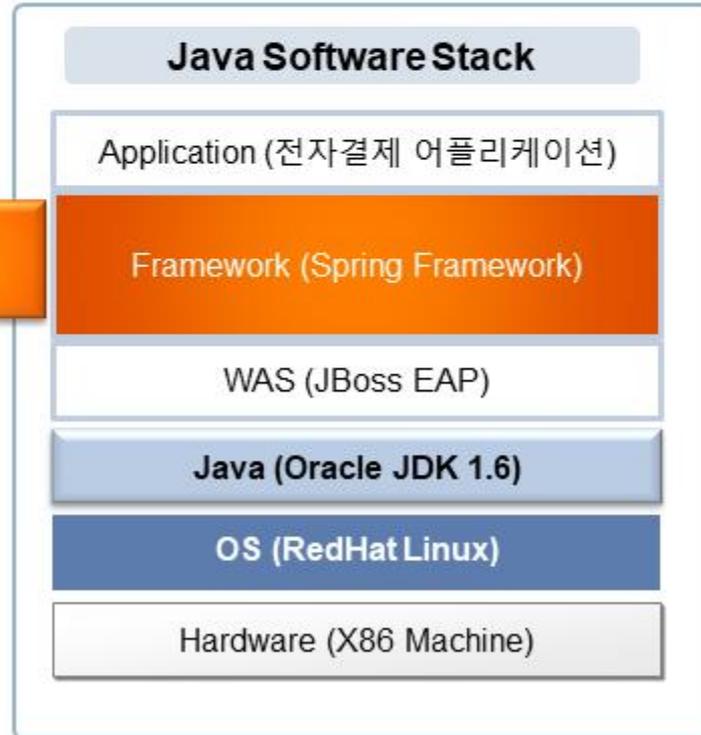
The Age of Framework



Source : http://raibledesigns.com/rd/entry/re_moving_from_spring_to

What is Framework ?

- 특정기술 또는 어플리케이션을 구성하는 기반구조로 SI 프로젝트에서 반복적으로 사용 되는 디자인 및
- 기반 소스 코드를 정의한 재활용 자산임
- **Spring , Hibernate, SEAM, iBatis,**



GS샵 - 스케일 아웃 용이한 공개SW로 고효율

GS SHOP



- 공개 SW 중심의 인프라 구축
- 유닉스 장비에서 X86 장비 중심으로 전환

스프린트 - 비즈니스 중심 IT 기회를 고갈시키는 비싼 미들웨어를 공개SW로 전환

Sprint



- WebLogic CPU 1,000개, WebSphere PVU 100,000개를 사용한 100개 이상을 JBoss EAP로 전환
- 연간 40 억원 이상 절감
- 미션 크리티컬 애플리케이션 환경 표준화 및 현대화

“독점형 미들웨어의 과도한 라이선스 및 유지 관리 비용을 지불하는 데 지친 Sprint는 비용의 획기적인 절감과 동시에 회사의 미션크리티컬한 비즈니스 애플리케이션에 향상된 유연성과 민첩성을 제공하는 Jboss로 전환

정부통합전산센터 - “공개S/W비중 50% 늘린다.”

Web	JBoss EWS
WAS	JBoss EAP
DBMS	Cubrid/Tibero/Altibas
가상화	RedHat Virtualization
OS (운영체제)	<ul style="list-style-type: none"> • Windows 2008 • RedHat Linux
하드웨어	X86 서버

클라우드 컴퓨팅

스마트전자정부

오픈 소스

예산 절감

LG전자 - 공개 SW를 활용한 Multi Platform 구현

LG Smart TV

- 신규시스템의 60% 이상이 공개SW기반 플랫폼으로 구현
- 기존 대비 최소 37%, 최대 70%의 Cost 절감

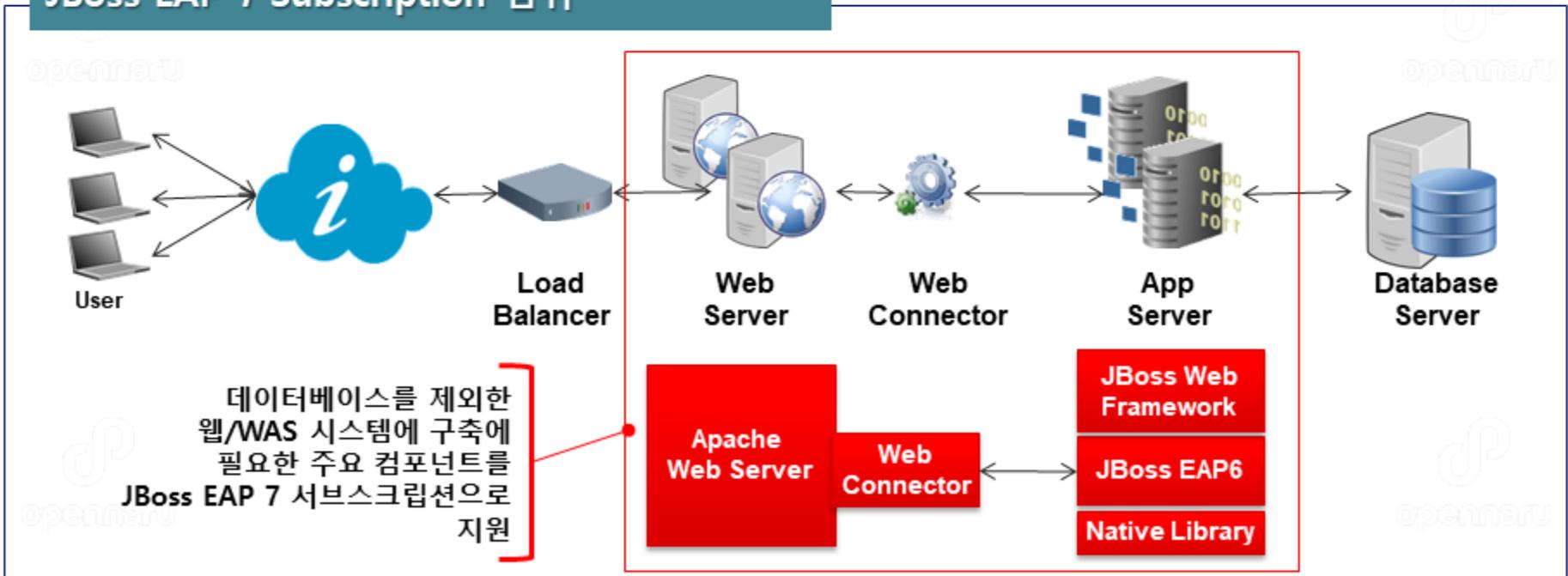
- 시스템의 중요도와 무관한 고비용의 Platform 활용
- 대형서버의 과다 비용 발생
→ 서버비용의 역전현상 (다수의 중형 서버 < 단일 대형 서버)
- 제조사 기술에 대한 종속현상 발생

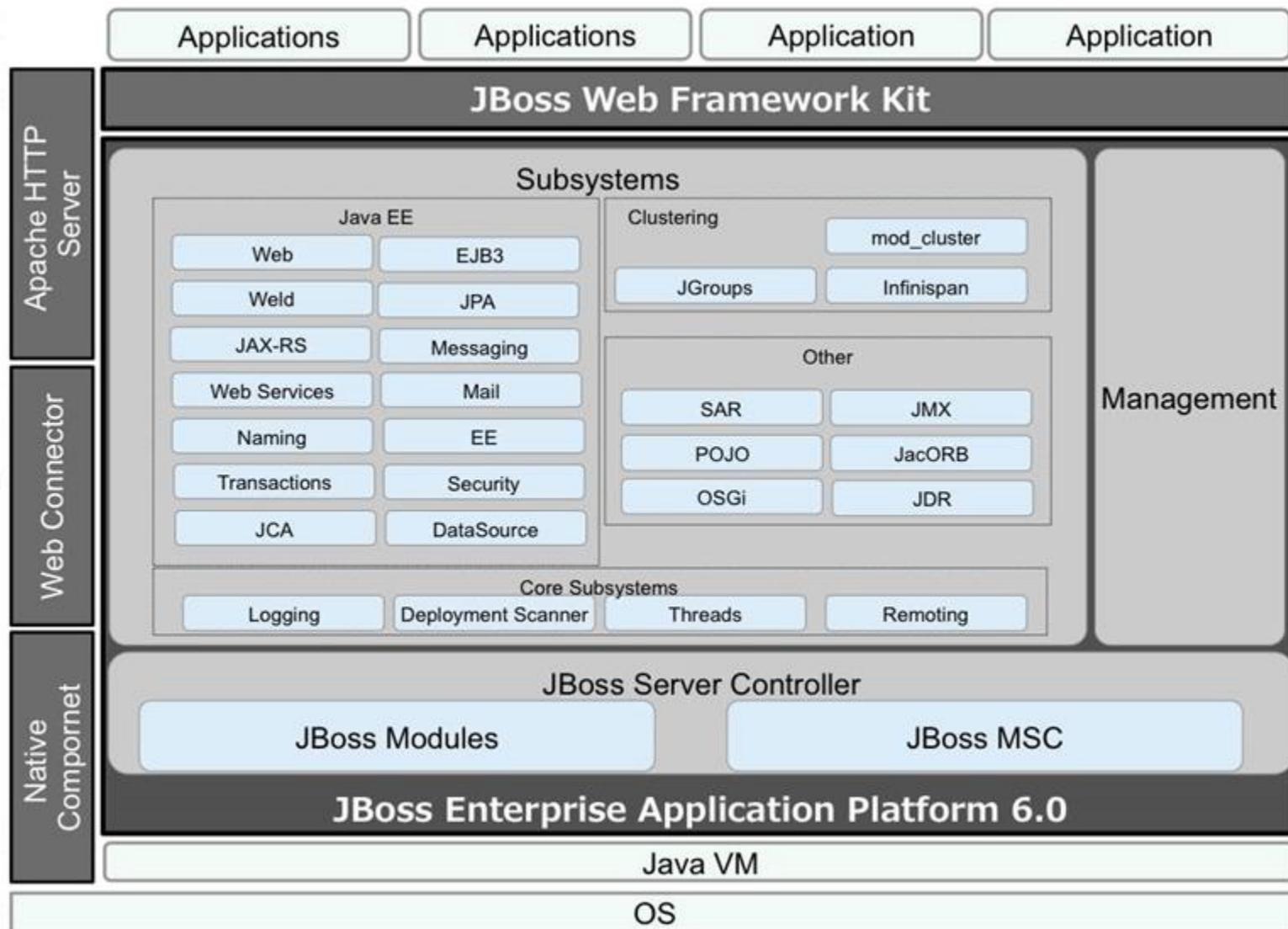
A close-up photograph of several lavender flower spikes. The central spike is in sharp focus, showing the dark purple buds and the emerging light purple flowers. The background is a soft, out-of-focus green, suggesting a garden setting. The text 'JBoss EAP 7 소개' is overlaid on the lower-left portion of the image.

JBoss EAP 7 소개

- Native 컴포넌트 - APR API를 이용한 Native 커넥터
- Web Connector - mod_jk, mod_cluster 등
- JBoss Core Service HTTPD Apache HTTP Server - 웹서버
- JBoss Web Framework Kit
 - 다양한 오픈소스 프레임워크 지원 - RichFaces, Snowdrop, Hibernate, Errai, TorqueBox, Seam, Spring, Apache Struts, Google Web Toolkit, Arquillian

JBoss EAP 7 Subscription 범위

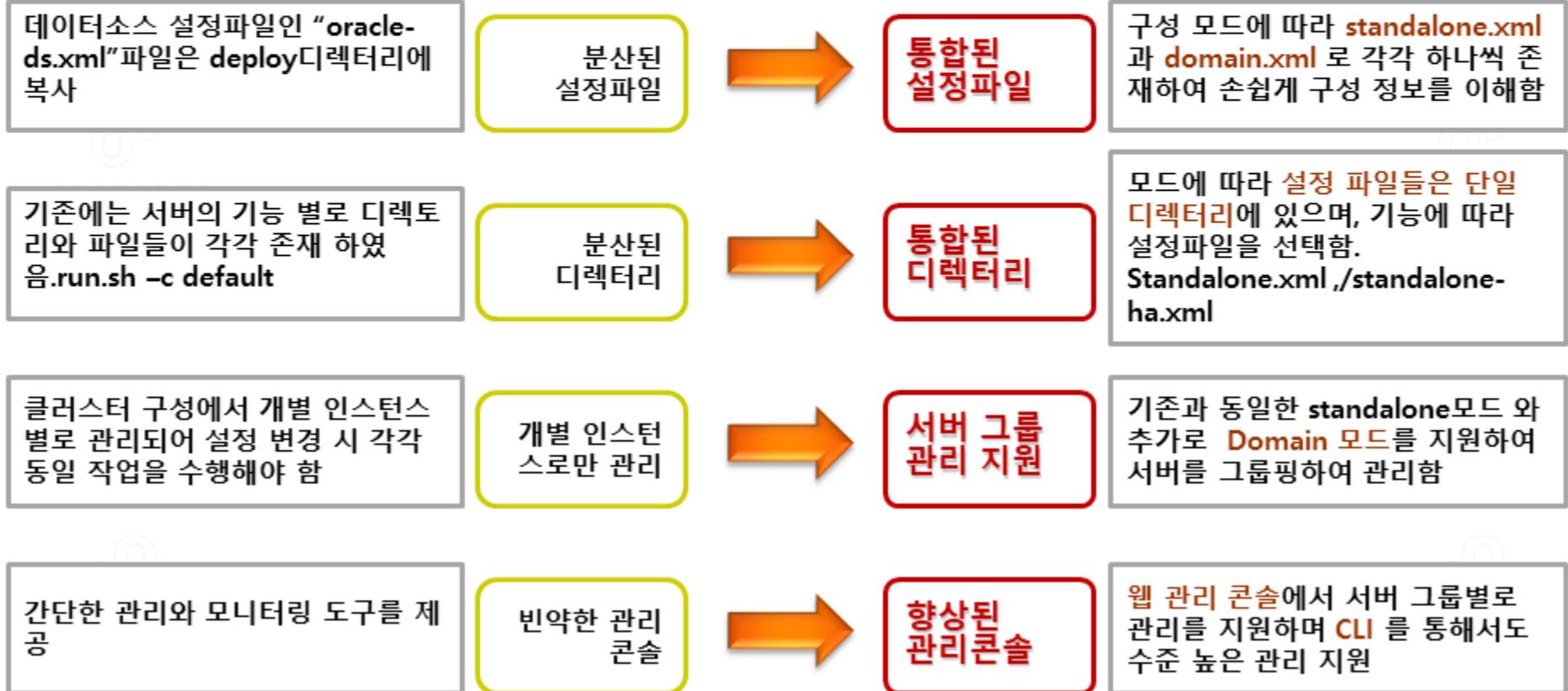




- 일관된 관리 인터페이스가 없어서 초보자가 이해하기 어렵고, 숙련자도 운영 실수를 유발하기 쉬운 구조
- 학습 유도나 관리 자동화

기존 WAS의 관리 및 설정 이슈

JBoss EAP 7



- 가상화/클라우드 환경에 적합한 가볍고 빠른 WAS
- 서브스크립션 방식의 비용 효율성

기존 WAS의 이슈

- 주요 WAS 구성 작업관련 **설정 파일 분산**
- 계층적 클래스로딩에 따른 **클래스 충돌**
- WAS 자체로 수 백 메가의 **많은 메모리 사용**
- 수 십 초 걸리는 **느린 부팅 타임**
- 관리 API** 제공 미흡
- WAS **Slimming** 어려움
- WAS 관리를 위해 **시스템 계정 요구**
- Standalone** 방식- **단일 인스턴스 관리 방식**
- 수 작업과 WAS **관리 자동화 기능 부재**

JBoss EAP 7

- 단일화된 **환경설정 파일 및 이력관리**
- 모듈형 클래스로딩** - 클래스충돌해결
- 수 십메가의 **낮은 메모리 사용**
- 4초 startup** 시간
- REST API/JSON** - 리모트 관리 API 제공
- 웹 기반 관리 콘솔**을 통한 WAS 관리
- 도메인모델**을 통한 서버 그룹 및 배포관리
- In-Memory Datagrid** 내장
- CLI 및 Batch Script**
- 빠르고 가벼운 WAS**

클라우드 레디
미들웨어



Public / Private Cloud
에서 가장 많이 사용되는
차세대 미들웨어



- **Java EE6 표준 지원**
- 최신 엔터프라이즈 자바 표준 기술을 이용한 어플리케이션 개발 지원
- 웹 프로파일에서 EJB3 Lite 지원



- **누구도 따라 올 수 없는 빠른 부팅 시간**
- 부팅 시 멀티 코어 지원과 병렬 클래스로딩
- 요청하지 않은 서비스는 사용되지 않음



- **개발 기간 단축**
- Spring, GWT, Struts, Maven 등 주요 오픈 소스 프레임워크에 대한 기술 지원
- 빠른 부팅시간을 통한 신속한 테스트



- **모듈형 클래스 로딩**
- 필요한 기능에 대하여 모듈형태의 패키지 구조를 적용
- 명시적인 클래스 의존성 정의



- **다이어트된 메모리 사용**
- 어플리케이션에서 필요한 서비스만 메모리에 로딩
- Jar 파일에 대한 인덱스 관리



- **인덱스를 통한 성능 향상**
- 클래스 정보와 Annotation 정보에 대하여 인덱스로 관리
- 인덱스된 정보는 캐시로 관리



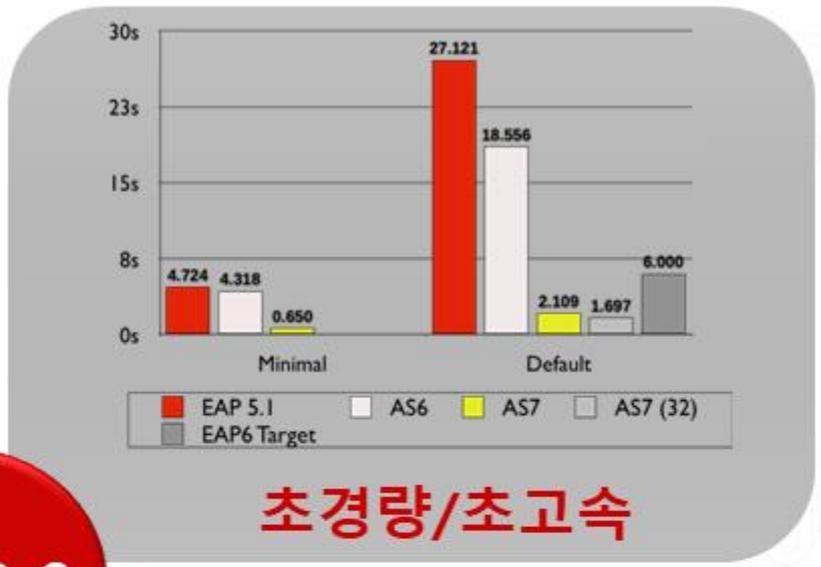
- **중앙 집중화된 설정 파일**
- EAP6 설정 정보들은 중앙화
- 변경된 설정 내용에 대하여 버전관리



- **CLI 를 통한 커맨드 형식의 관리**
- 중앙화된 설정방법들을 통하여 관리 작업을 통합
- Batch 스크립트를 이용하여 주요 작업에 대한 자동화와 재사용

JAVA EE6
Spring
HIBERNATE
GWT
Apache
maven

OSS 프레임워크 지원



EAP 6

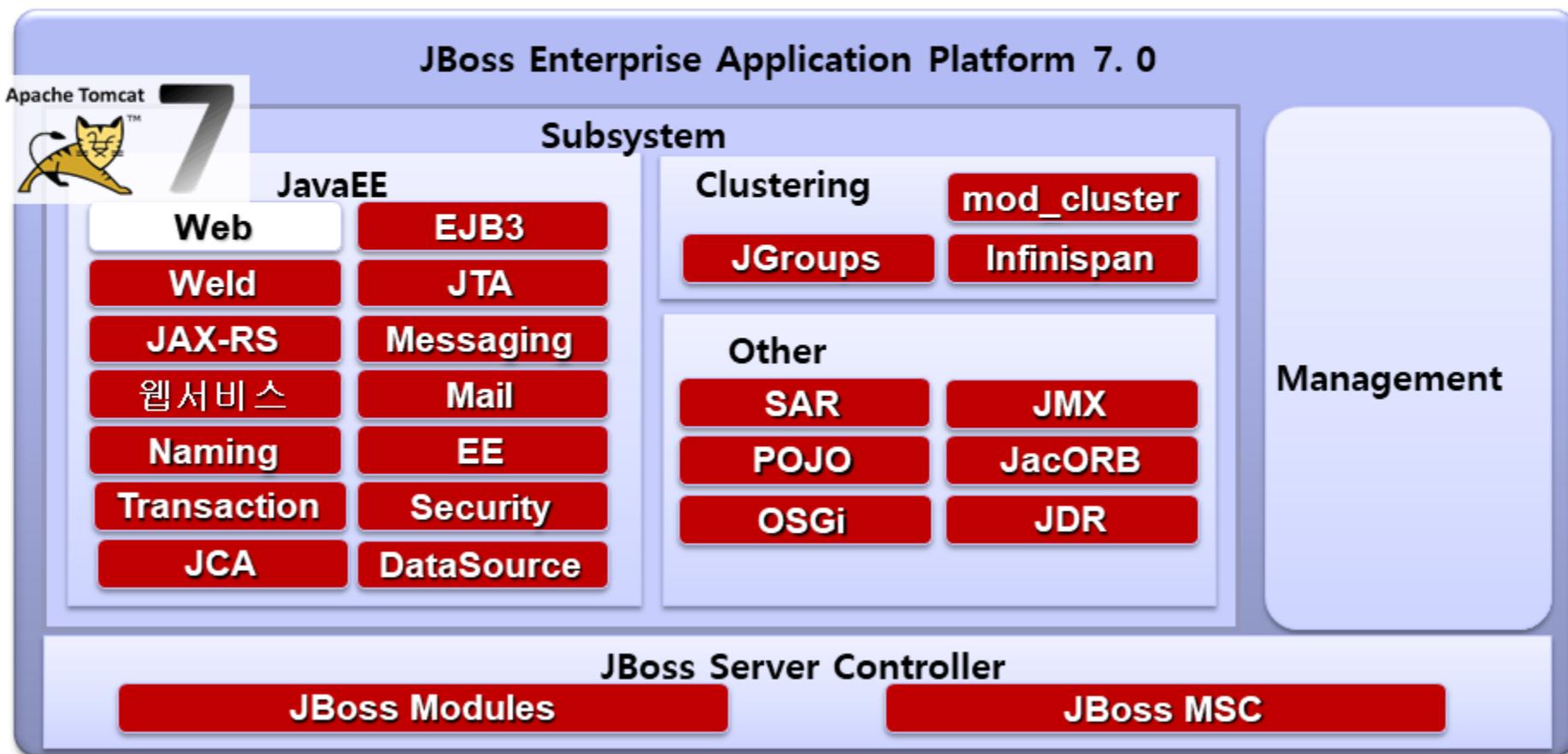
풍부한 관리 인터페이스

JBoss OPERATIONS NETWORK
MANAGE

OPENSIFT

데이터그리드 / 클라우드

- EAP 7에서는 EAP 6부터 새롭게 모듈 시스템(JBoss Modules)을 도입
- MSC(Modular Service Container)로 불리는 컨테이너가 모듈을 관리(로드)
- 클래스(class)는 module 단위로 로드 되어 MSC 에 의한 module 기반의 클래스 로딩 (비계층형)



JBoss EAP 7 Subsystem : Subsystem 리스트-1

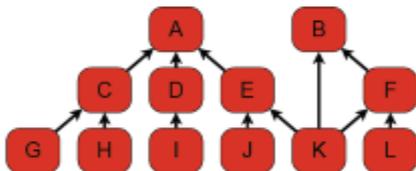
subsystem	extension	xmlns
logging	org.jboss.as.logging	urn:jboss:domain:logging:1.1
datasources	org.jboss.as.connector	urn:jboss:domain:datasources:1.0
deployment-scanner	org.jboss.as.deployment-scanner	urn:jboss:domain:deployment-scanner:1.0
ee	org.jboss.as.ee	urn:jboss:domain:ee:1.0
ejb3	org.jboss.as.ejb3	urn:jboss:domain:ejb3:1.1
infinispan	org.jboss.as.clustering.infinispan	urn:jboss:domain:infinispan:1.0
jaorb	org.jboss.as.jaorb	urn:jboss:domain:jaorb:1.0
jaxrs	org.jboss.as.jaxrs	urn:jboss:domain:jaxrs:1.0
jca	org.jboss.as.connector	urn:jboss:domain:jca:1.0
jgroups	org.jboss.as.clustering.jgroups	urn:jboss:domain:jgroups:1.0
jmx	org.jboss.as.jmx	urn:jboss:domain:jmx:1.0
jpa	org.jboss.as.jpa	urn:jboss:domain:jpa:1.0
messaging	org.jboss.as.messaging	urn:jboss:domain:messaging:1.0
modcluster	org.jboss.as.modcluster	urn:jboss:domain:modcluster:1.0
naming	org.jboss.as.naming	urn:jboss:domain:naming:1.0
osgi	org.jboss.as.osgi	urn:jboss:domain:osgi:1.0
remoting	org.jboss.as.remoting	urn:jboss:domain:remoting:1.0

subsystem	extension	xmlns
resource-adapters	org.jboss.as.connector	urn:jboss:domain:resource-adapters:1.0
sar	org.jboss.as.sar	urn:jboss:domain:sar:1.0
security	org.jboss.as.security	urn:jboss:domain:security:1.0
threads	org.jboss.as.threads	urn:jboss:domain:threads:1.0
transaction	org.jboss.as.transactions	urn:jboss:domain:transactions:1.0
web	org.jboss.as.web	urn:jboss:domain:web:1.0
webservices	org.jboss.as.webservices	urn:jboss:domain:webservices:1.0
weld	org.jboss.as.weld	urn:jboss:domain:weld:1.0

직렬형 로딩 방식

- 단일 스레드 방식으로 서비스 로딩
- CPU 를 잘 활용하지 못하는 구조
- 부팅 시 모든 서비스를 로딩하는 구조

Serial DAG Execution (Traditional Approach)

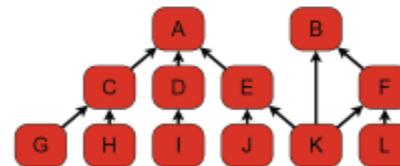


CPU Core	CPU Core	CPU Core	CPU Core
A	Zzz	Zzz	Zzz
B	Zzz	Zzz	Zzz
C	Zzz	Zzz	Zzz
D	Zzz	Zzz	Zzz
E	Zzz	Zzz	Zzz
F	Zzz	Zzz	Zzz
G	Zzz	Zzz	Zzz
H	Zzz	Zzz	Zzz
I	Zzz	Zzz	Zzz
J	Zzz	Zzz	Zzz
K	Zzz	Zzz	Zzz
L	Zzz	Zzz	Zzz

병렬형 로딩 방식

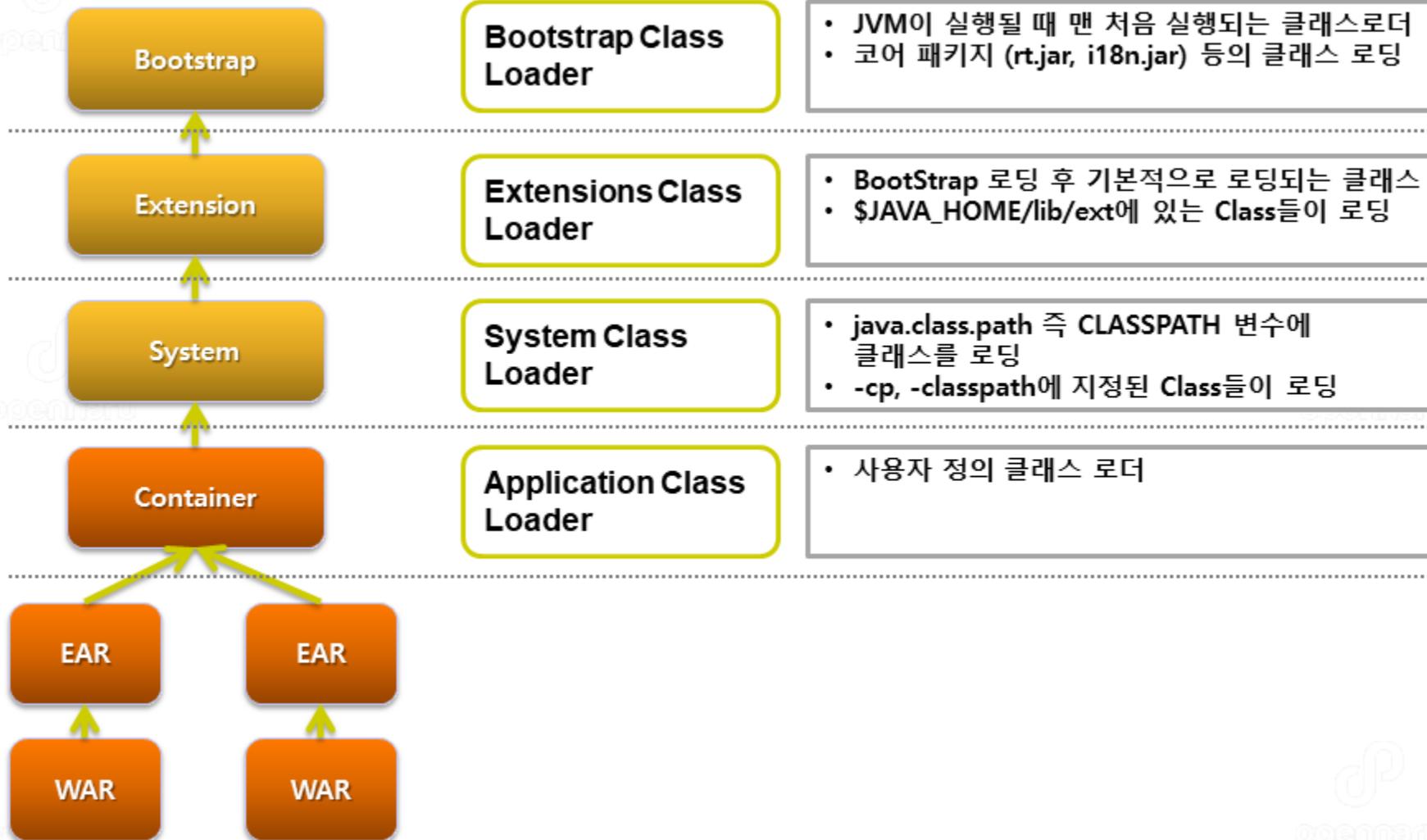
- 멀티 스레드 방식으로 모듈 로딩
- 멀티 코어 지원을 통한 신속한 프로세스
- 사용되는 모듈만 로딩 되는 구조
- 모듈에 대한 의존성 관리를 Concurrent DAG 적용

Concurrent DAG Execution (MSC / EAP Approach)



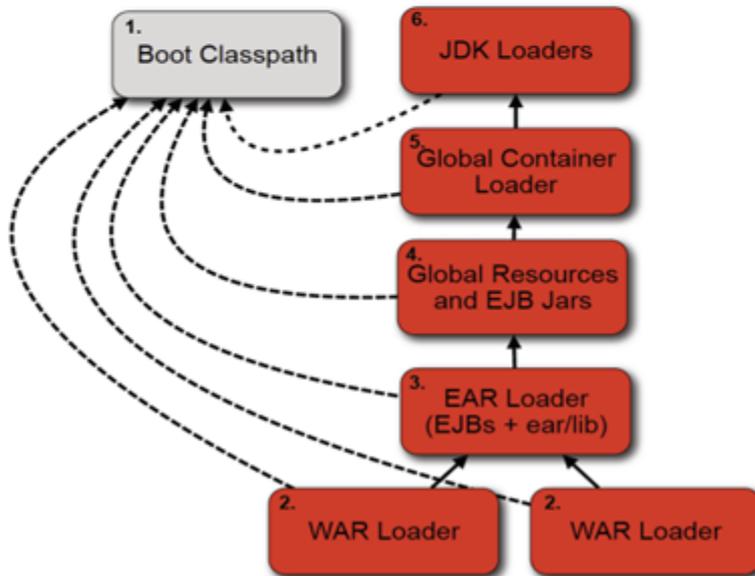
CPU Core	CPU Core	CPU Core	CPU Core
A	B	Zzz	Zzz
C	D	E	F
G	H	I	J
K	L	Zzz	Zzz

* DAG = Directed Acyclic Graph (방향성 비순환적 그래프)



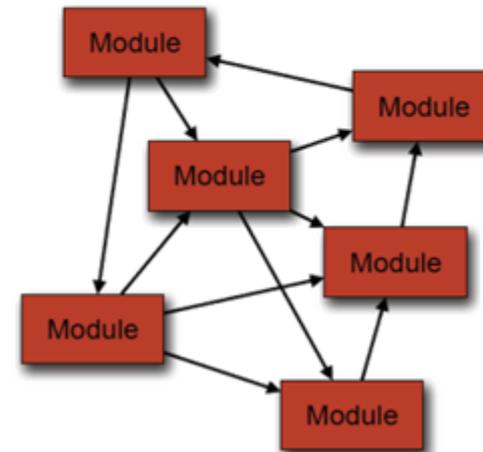
계층형 클래스로더

- 중복 배포
- 로드 순서에 의한 교착 상태 발생
- 복잡/클래스 검색이 늦음
- 중복 배포에 의한 오류 발생
- 클래스 공유
- 문제를 회피하기 위한 구조가 더 복잡도를 높여 악순환

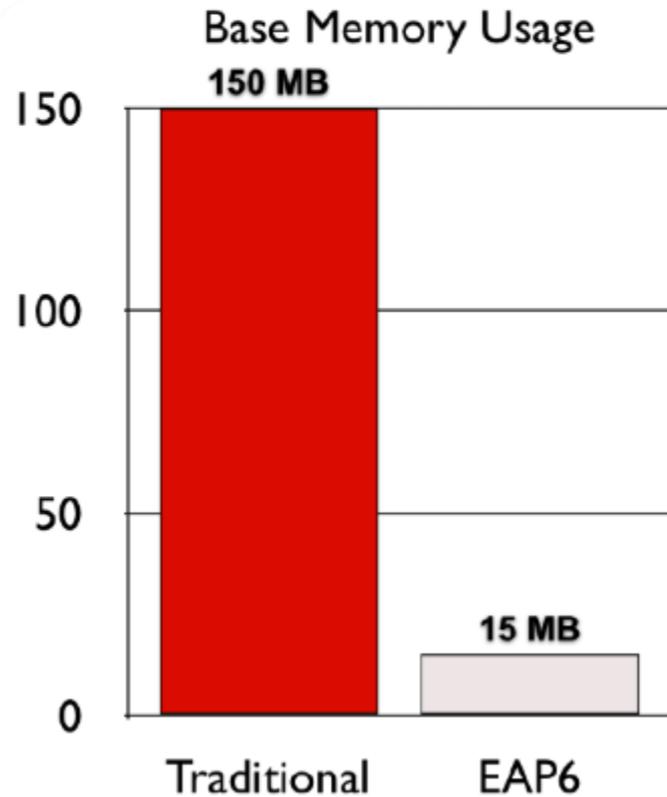
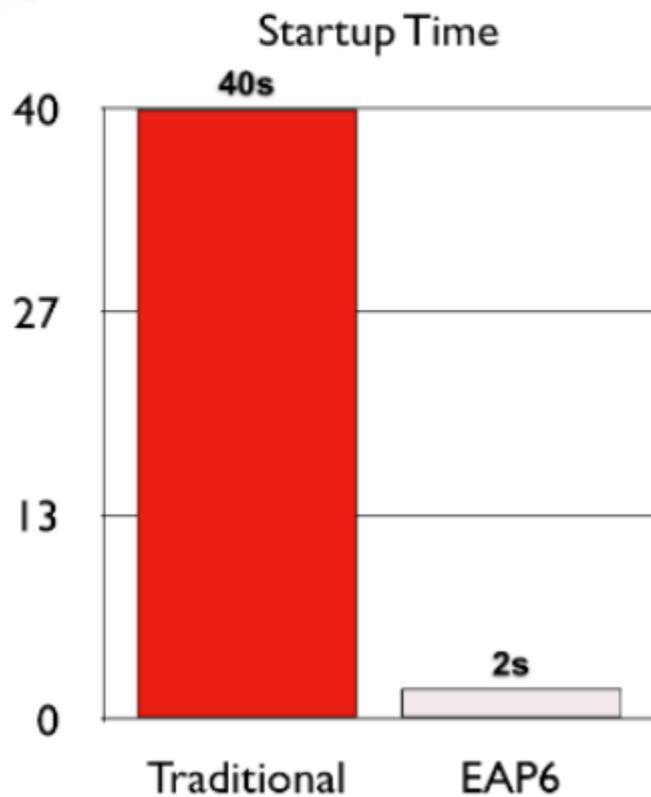


모듈형 클래스로더

- 계층형 클래스 로더의 문제점 해결
- 모듈 하나에 대해서 하나의 클래스로더
- 각 모듈은 런타임으로 필요로 하는 모듈의 의존성을 정의
- 계층형이 아닌 그래프 구조
- 「클래스 패스」는 사라짐
- 단순하여 초고속

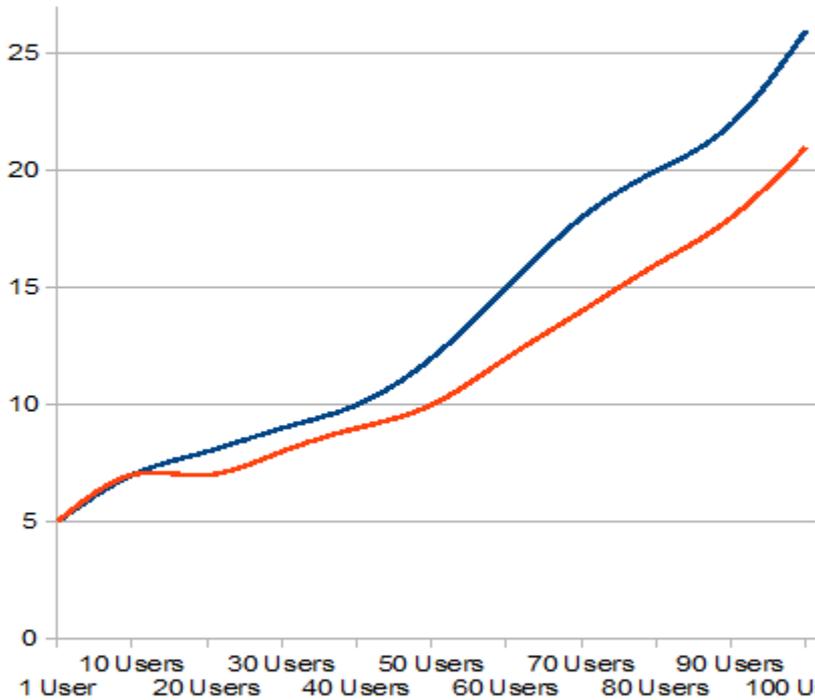


클라우드 운영 환경에서 신속한 구성 지원
빠르고 가벼운 WAS

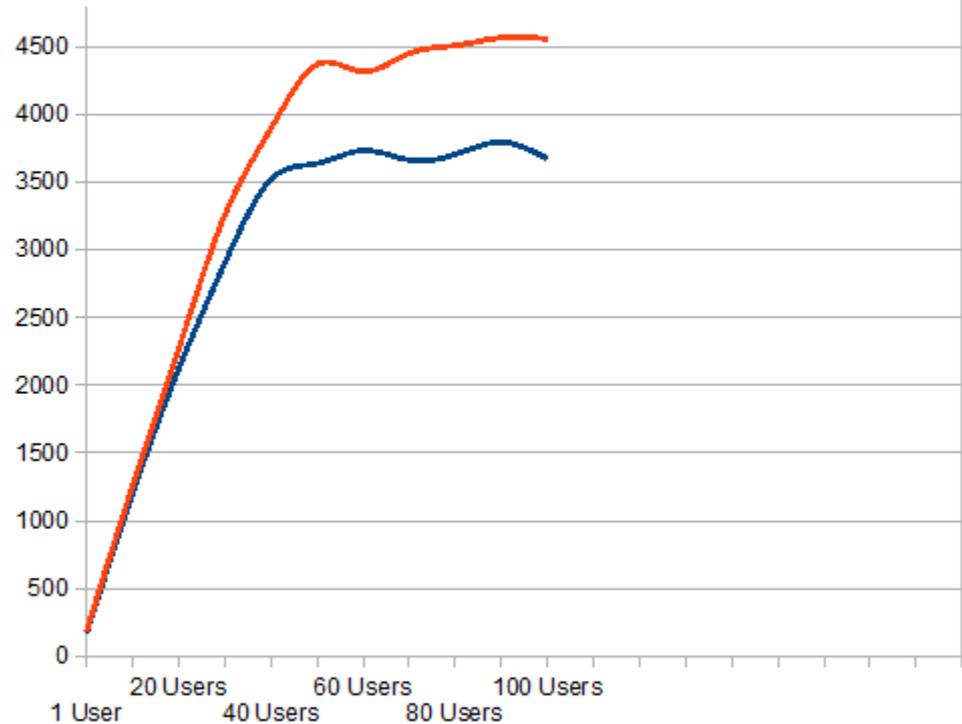


— EAP 5.1.1
— AS 7.1.0.Alpha 2

응답시간



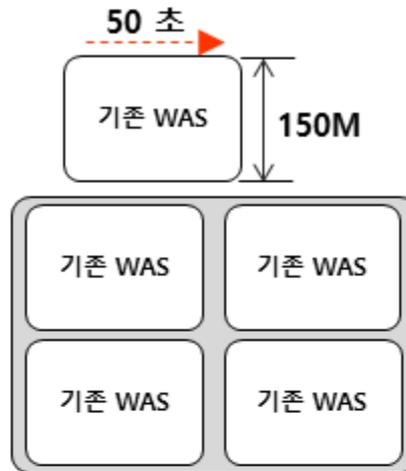
throughput



EAP5 대비 20% 성능 향상

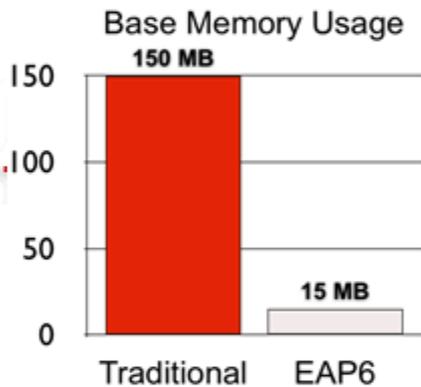


Unix 장비 중심



고비용 고사양 Unix 장비

- Scale Up
- 느린 부팅 시간과 많은 메모리 사용
 - 부팅 시 많은 시간 소요
 - WAS 관리 작업 시 많은 시간 소요
- Unix 장비 - 고비용 고사양 장비
- Java EE 스펙 지원에 필요한 모든 클래스 로딩



4개 인스턴스 / Rolling Restart 시 3분 33초



Commodity X86 장비

- Scale Out
- 초경량 / 초고속 어플리케이션 서버
 - 몇 십메가 / 몇 십초
 - GC 횟수 감소에 의한 성능 향상
 - 빠른 시작/정지
- 마이크로 인스턴스 관리 / 리소스 사용 효율화
- Blade 급 서버 추가
- 단일 파일 설정으로 맞춤형 프로파일 작성

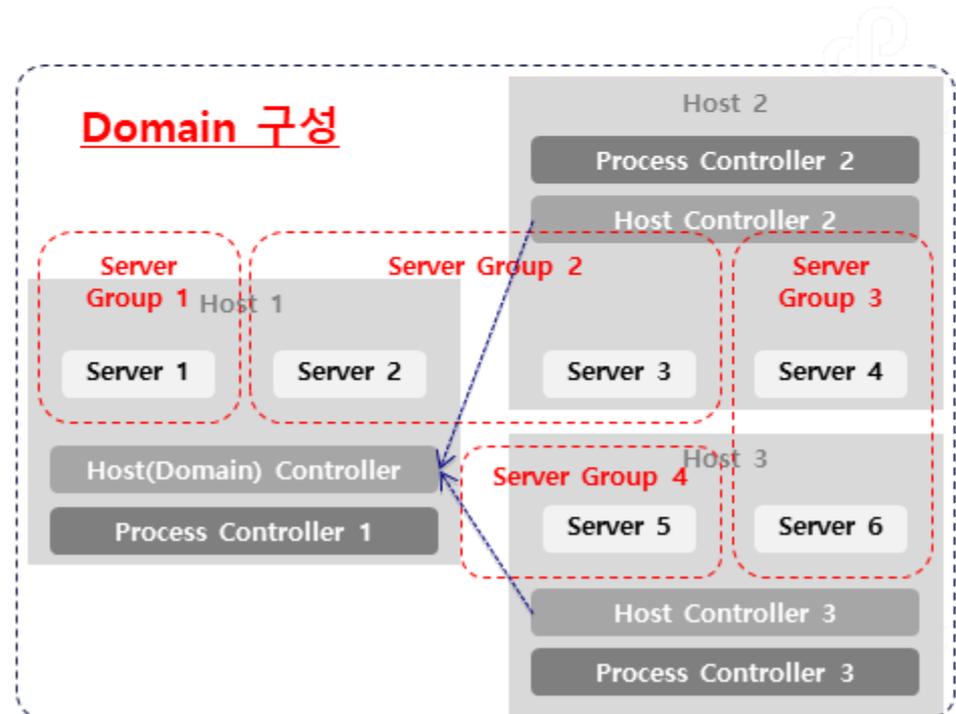


X86 장비 중심

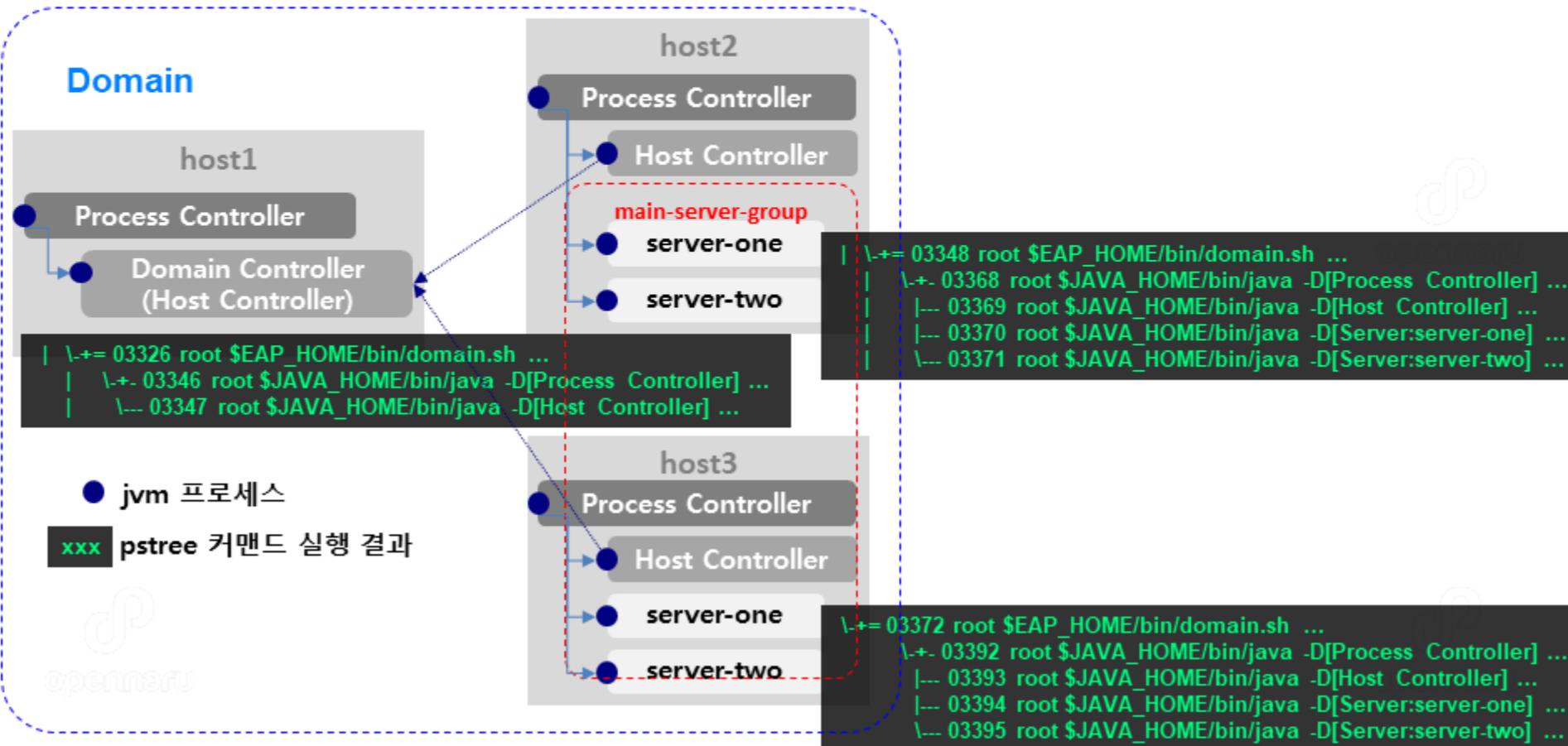
20개 인스턴스 / Rolling Restart 시 1분 33초

- Standalone 구성
- ✓ 단일 서버 프로세스 구성
- ✓ 여러 서버 인스턴스를 독립적으로 가동
- ✓ 동일한 어플리케이션을 복수의 서버 인스턴스에 배포하는 경우, 배포 작업을 개개의 server 인스턴스에 대해서 실시

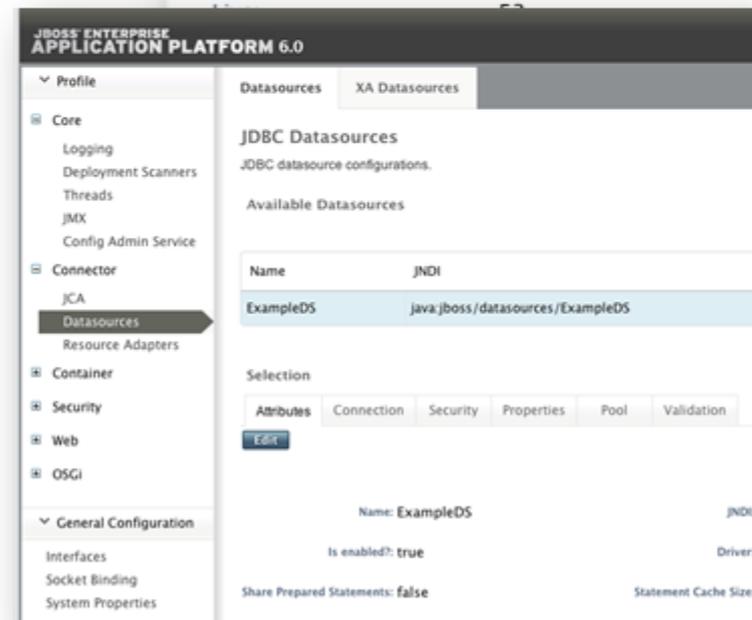
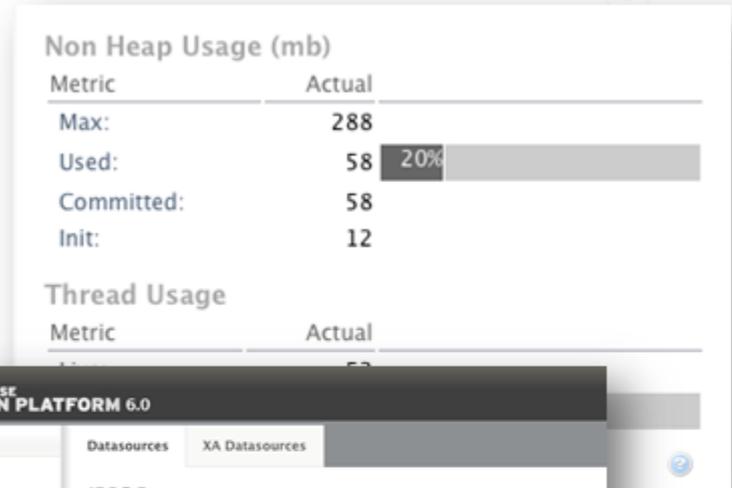
- Domain 구성:
- ✓ 모든 서버 인스턴스는 동일한 정책으로 관리
- ✓ domain 컨트롤러에 의해 서버 인스턴스들이 관리
- ✓ domain 는 물리/가상 머신에 걸쳐 확장 가능
- ✓ 각각의 host 는 host 컨트롤러 프로세스의 제어
- ✓ domain 중의 하나의 host 컨트롤러는 domain 컨트롤러로서 동작
- ✓ host 컨트롤러는 Host의 server 인스턴스 제어 및 domain 컨트롤러와 연결



- 프로세스 컨트롤러, 호스트 (도메인) 컨트롤러, 서버 인스턴스는 각각 독립적인 JVM 프로세스
- 호스트 컨트롤러, 서버 프로세스는 프로세스 컨트롤러 프로세스에 의해 기동
- 호스트 컨트롤러 중 도메인 컨트롤러로 정의된 것이 도메인 컨트롤러로 동작



- 가볍고 빠른 웹어플리케이션 형태
- not XML
- 거의 모든 설정 지원
- Standalone / Domain 구성
- 접속
- 변경내용 저장
- 모니터링



- 시스템의 자원을 트리 구조로 액세스

- 주소는 Key/Value 구성

- ✓ Standalone 구성- 예) /subsystem=web/connector=http

- ✓ Domain 구성- 예) /profile=default/subsystem=web/connector=http

- 자원에 대해서 작업과 파라미터를 지정

- cd 로 이동, ls 목록 출력, tab 키로 다음 단계에서 실행 가능한 명령어 제안

예, Standalone 모드에서 모든 시스템 프로퍼티 출력 - 전체 출력

```
[standalone@IP ADDRESS:9990 /]  
/core-service=platform-mbean/type=runtime:read-attribute(name=system-properties)
```

주소

operation
(: 시작)

파라미터
(() 닫기)

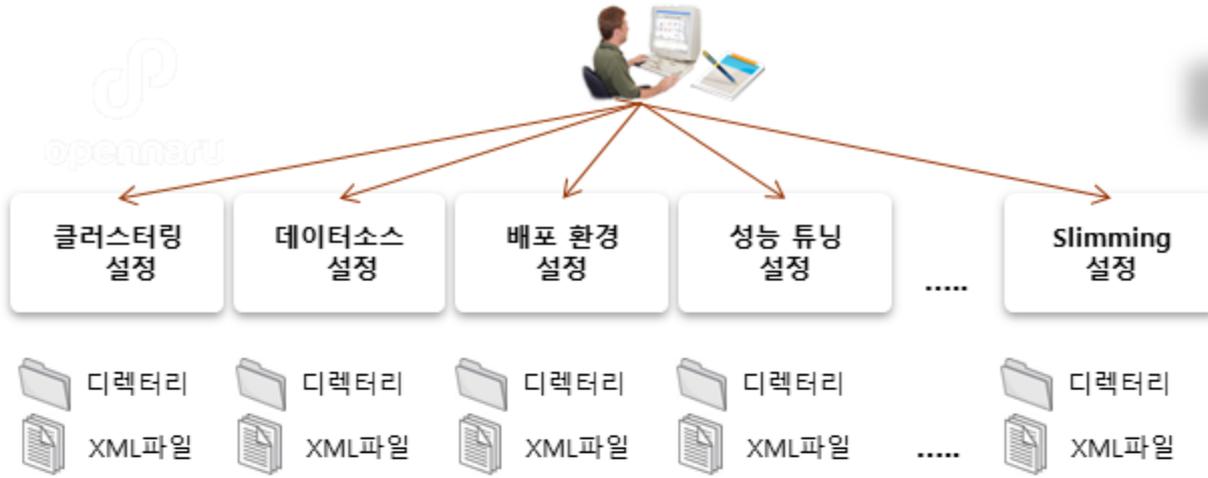
예, Standalone 모드에서 모든 시스템 프로퍼티 출력 - 단계별

```
[standalone@localhost:9990 /] cd core-service=platform-mbean  
[standalone@localhost:9990 core-service=platform-mbean] cd type=runtime  
[standalone@localhost:9990 type=runtime] :read-attribute(name=system-properties)
```

주소

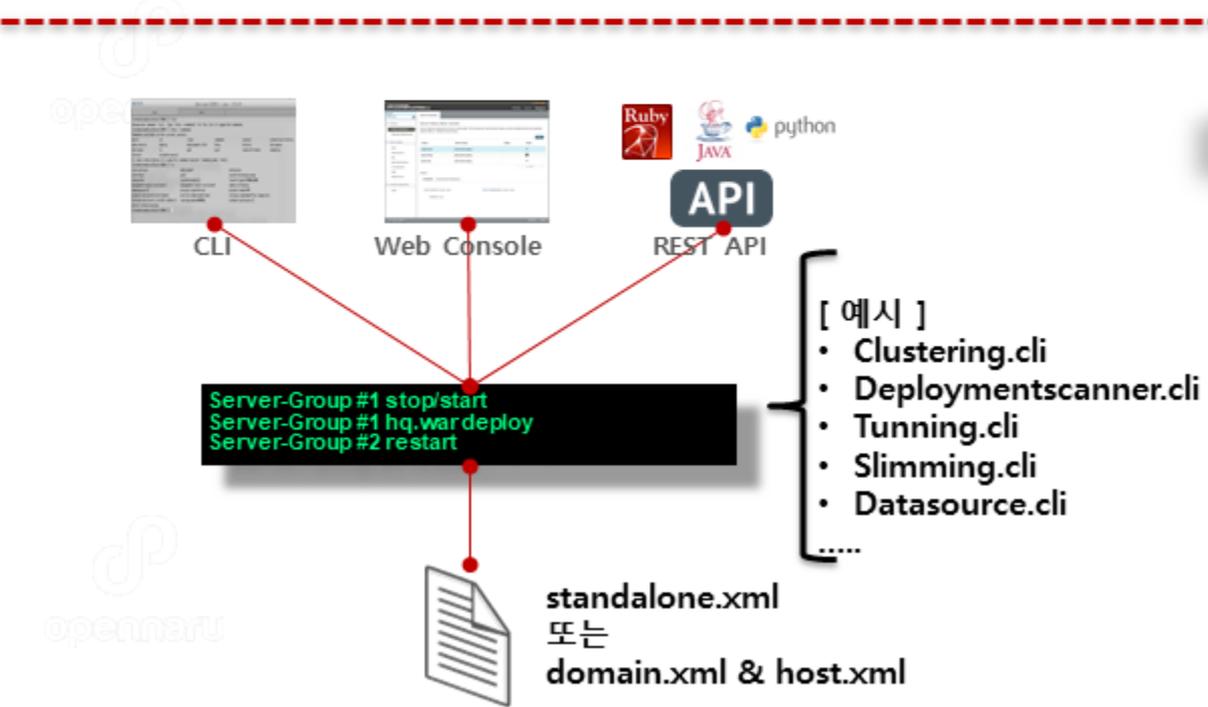
operation

파라미터



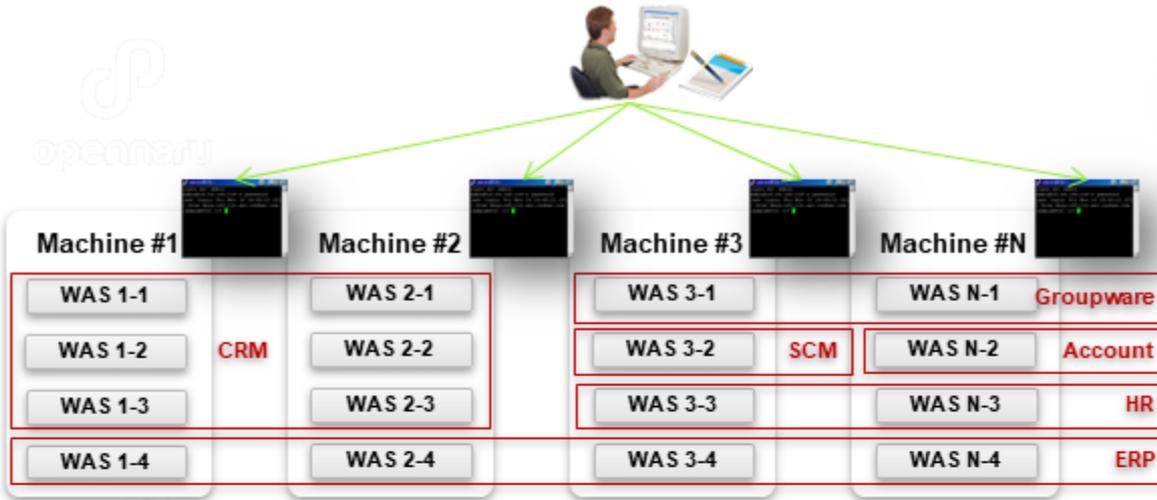
분산된 설정 파일들과 수작업

- 주요 WAS 관련 작업마다 개별 디렉터리와 파일들을 수작업으로 편집해야 함
- 전문 지식이나 경험이 필요함
- 수작업에 의한 운영자 오류에 주의해야 함

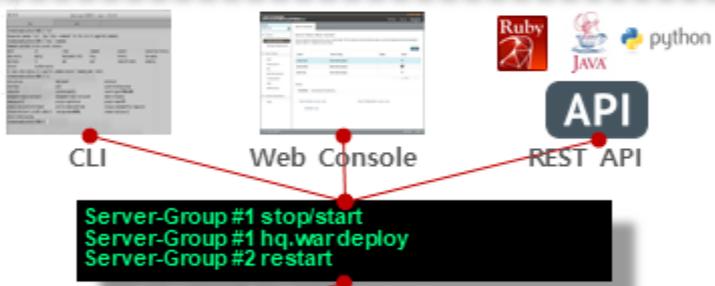


일원화된 관리 방법과 자동화

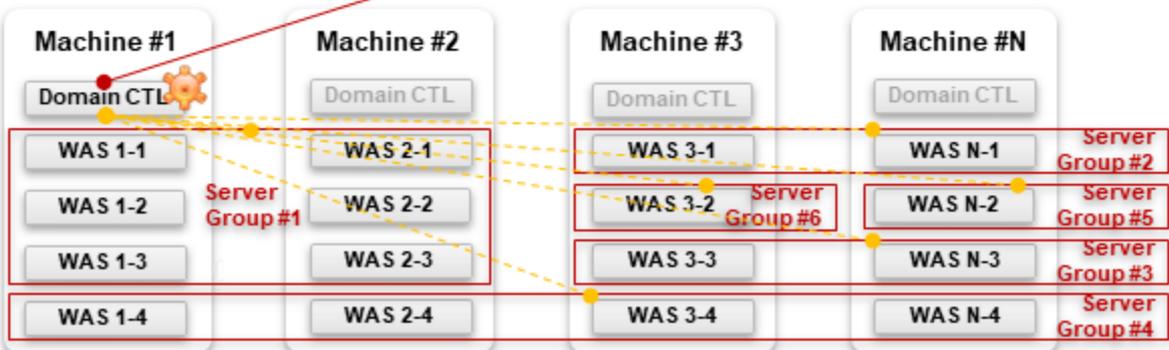
- 구성 방법 (standalone 또는 domain) 에 따른 일원화 되고 통합된 xml 파일 관리
 - xml 파일에 대한 history 생성
 - Single Log On
- CLI 스크립트를 통한 자동화와 템플릿
 - WAS 관리에 필요한 주요 Task 에 대하여 cli 스크립트를 통한 설정



- ### WAS 관리의 현실
- System 관리자와 WAS 관리자 분리의 어려움
 - System ID/PWD 를 WAS 관리자에게 노출
 - 수작업에 의한 Configuration 관리
 - Human Error
 - 로그 파일 분산
 - 서버 환경 동기화의 어려움
 - 동일 작업을 N 번 반복



- ### 현대화된 WAS 관리 방안
- 중앙에서 서버 관리 일원화
 - 서버 그룹 단위 관리
 - 서버 그룹 단위의 start /stop
 - 서버 그룹 단위의 Application 배포
 - 통합 모니터링
 - Shell 이 아닌 웹콘솔이나 CLI 도구를 통한 관리
 - WAS 내부 구조를 노출하지 않음
 - 외부 도구를 위한 RESTful API 제공
 - 다양한 도구를 통하여 WAS 관리 자동화
 - 클라우드 환경에 맞는 구성 관리
 - 통합된 설정 정보들



EAP 7 에서의 배포 방법은 아래와 같이 4가지



웹 기반 관리 콘솔

- 웹 기반 관리 콘솔로부터 어플리케이션 업로드 후 배포



CLI

- 스크립트 기반 deploy / undeploy 커맨드 제공



Deployment Scanner

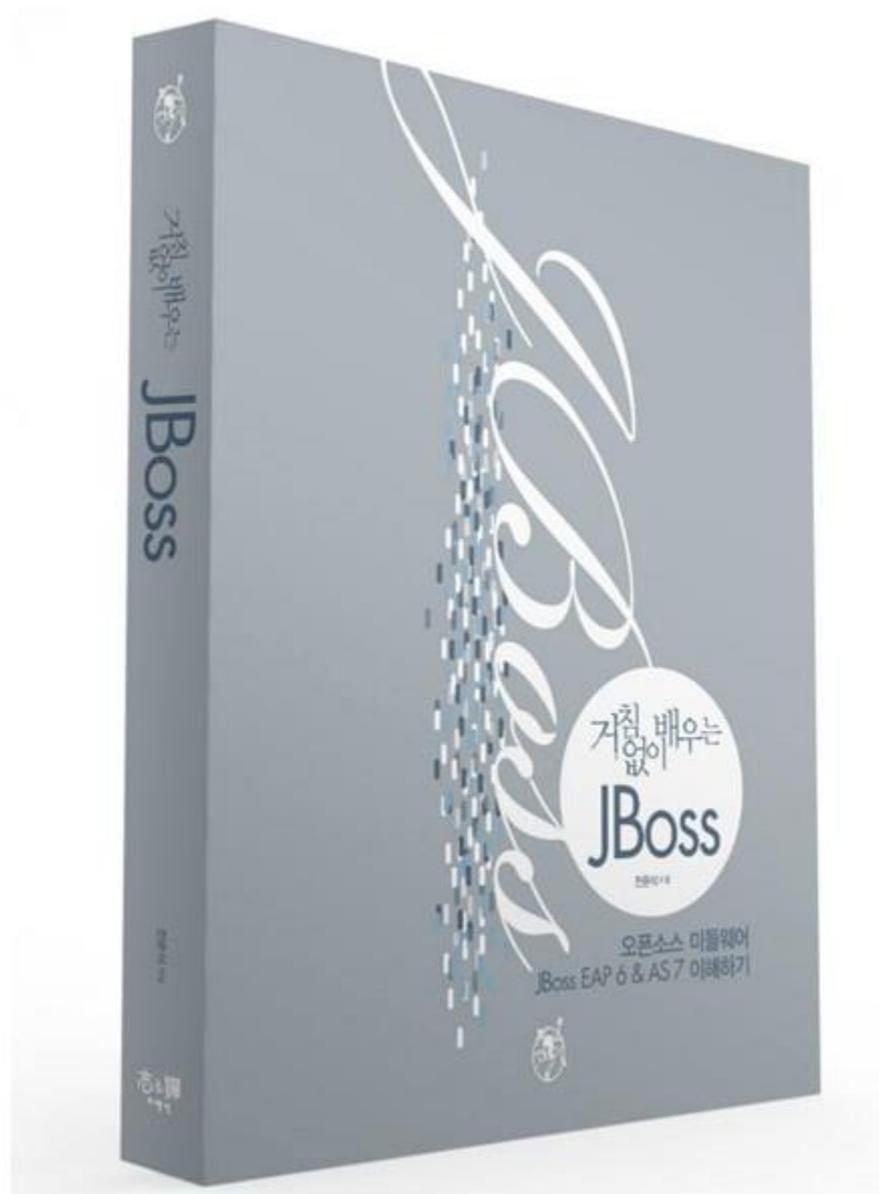
- JBoss EAP5 와 동일한 기능 (Domain 모드에서 지원하지 않음)



Maven 형태 배포

- Maven 커맨드에 의한 Deploy / Undeploy

다양한 배포 방법 과 도메인 모드에서 배포 지원





제품이나 서비스에 관한 문의

콜 센터 : 02-469-5426 (휴대폰 : 010-2243-3394)

전자메일 : sales@openmaru.com