

J2EE 어플리케이션 문제 해결 방법론

차의중
IBM Korea
WebSphere Field Technical Sales Support
ejcha@kr.ibm.com

선결 조건

□ 침착하라

- 심호흡
- Operating은 천천히, 또박 또박

□ 혼자 해결하려 하지 마라

- 물어보라
- 찾아보라
- 확인하라

□ 필요한 최소한의 인원만을 곁에 두어라

- 파트별 엔지니어, 어플리케이션 개발자

□ 메모하라

- 시간, 행위 (자세히)

대안의 준비

- 기술적 해결만이 최선이 아니다
 - 목표의 선회, 업무적 해결
- 넓게 보라
 - 문제는 어디서든 일어날 수 있고, 누구든 일으킬 수 있다
 - 정공법만이 최선이 아니다
- 단계별 한계 시간과 향후 전략을 미리 정해두라

문제의 구체화

□ 기술적 담당자

관리자

- 어플리케이션, 시스템, 데이터베이스...

개발자

- 설계, 코딩...

□ 업무적 담당자

- 전체 책임자, 어플리케이션 책임자, 현업 담당자

□ 지원 담당자

솔루션 판매자, 컨설턴트

- 영업 대표, 엔지니어...

*** 그 문제에 대한 담당자 지정 필요 ***

□ ex)느려진다

어디서 측정했는가, 어느 구간을 측정했나, 월로 측정했는가
얼마나 느린가 (평소 X 초, 문제시 Y초)

□ Ex)응답이 없다

뭐가 응답이 없는가 (특정 서비스, 어플리케이션, 서버, 데이터베이스, 시스템 전체)

응답이 아예 없는건가, 너무 느린건가

□ Ex)죽는다

뭐가 죽는가 (OS/프로세스/웹서버/JVM/어플리케이션)

어떻게 하고 죽는가 (로그, Core dump, 특이증상 없음)

□ Ex) 이상하게 동작한다

정상일 때는 어떤가 / 지금은 어떤가

화면이 전혀 안 보이나

다른 사람의 데이터가 나오나

이상하게 가공된 데이터가 나오나

글자나 이미지가 깨졌나

어느 부분이 이상한가

전체가 이상한가 / 특정 부분이 이상한가

- 언제 발생했는가
- 얼마나 자주 발생하는가
- 그 때만 되면 발생하는가
- 증상이 시간에 따라 변화하는가, 어떻게 변화하는가
- 다른 날과 비교해서 다른 점이 있는가
 - 코드, 설정, 환경, 사용자 수, 특정 작업의 호출수
 - 사용자의 성향, 데이터의 양...

□ 뭘 어떻게 했을 때 발생하는가

그렇게 하기만 하면 발생하는가

□ 상황이 변화 되었나

어떤 상황이 변화되었나

- 사용자 수, 특정 작업의 호출 빈도, 사용자의 성향, 데이터의 양

...

□ 손땀나

누가 손땀나

뭘 손땀나

- 코드, 설정, 환경, 패치...

어떻게 손땀나

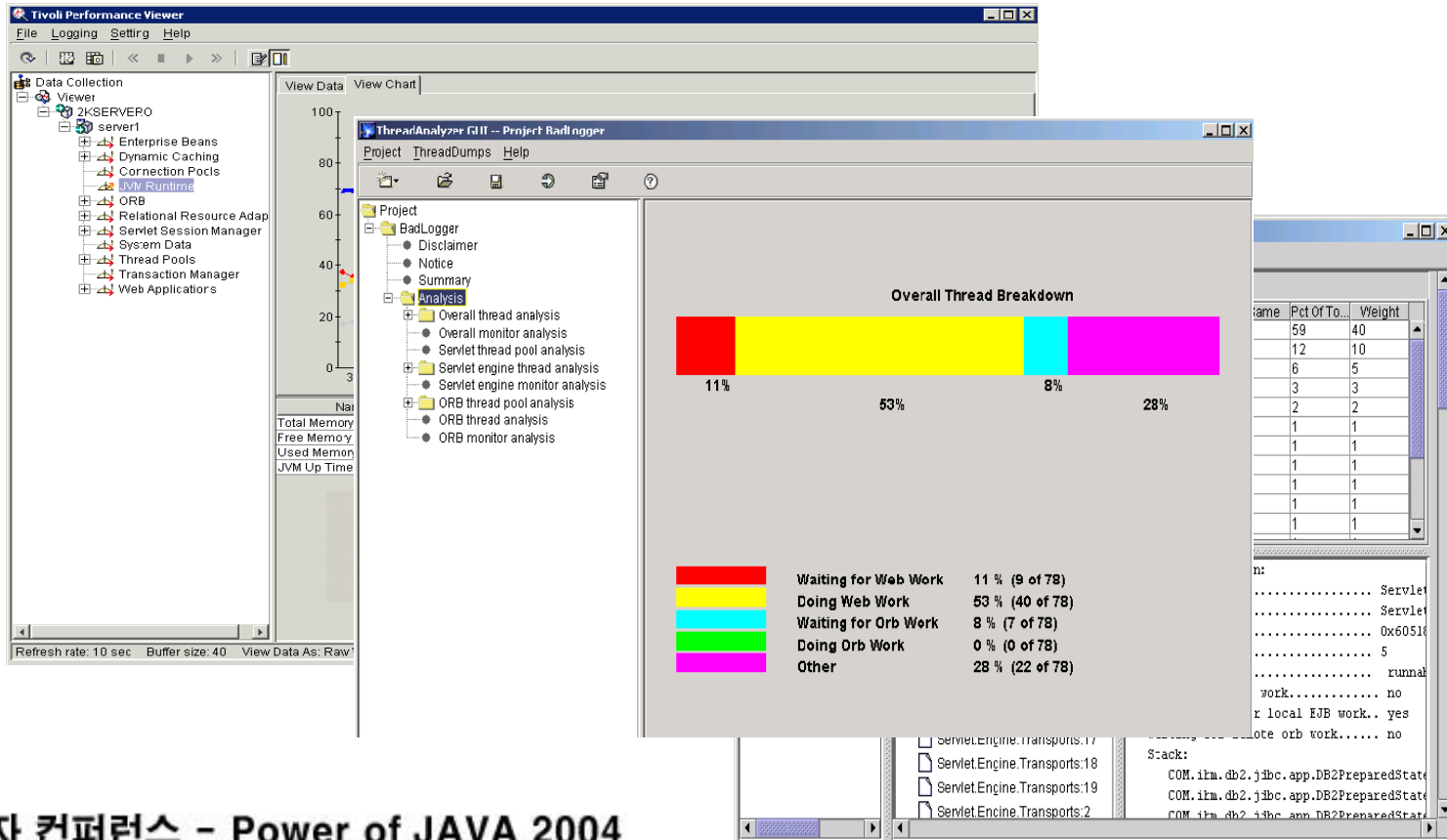
- 추가, 삭제, 변경

로깅 및 모니터링

- 장애 전부터 이미 시작되었어야 한다.
 - 제품 Install 시부터 시작되어야 한다.
- WAS 뿐만 아니라 기타 자원까지 로깅/모니터링하라
 - OS, Web Server, DB, Network, Firewall
- 충분히 많은 양을 로깅/모니터링하라
 - Stress Test시에 Disk Busy 가 심하지 않을 정도
- 장애 시점 및 그를 약간 전후한 시간대의 모든 자원을 의심하라

- JVM, 컨테이너, OS내 자원의 운용 상황, 시간적 변동 상황, 다양한 분석
- 모듈별 응답 시간, 스레드 상태

Ex) WebSphere



- 장애 상황, 운영 상황, 로직 흐름, 장애 지점, 장애 내용, 기타 런타임 환경의 상세 내용
 - Ex) WebSphere
 - SystemOut.log, SystemError.log, native.log, Access_log, error_log...
 - Core, java core <= 장애 시점의 메모리, 스레드 상태
 - <http://www.javaservice.net>

□ OS 레벨 자원 운용 상황

-topas, glance, vmstat

-특히, Memory 운용 상황은 OS 레벨에서의 측정치와 JVM 내부의 실제 운영치가 다르므로 반드시 JVM 레벨에서 모니터링 필요

□ 특정 포트간 커넥션 수 실시간 모니터링

-mon.sh (이원영)

□ 현재 구동중인 웹 페이지들 보기

-requestmon.jsp

□ JVM Heap 모니터링

-mem.jsp, m.jsp (이원영)

<http://www.javaservice.net>

필수 지식들

□ 컴포넌트?

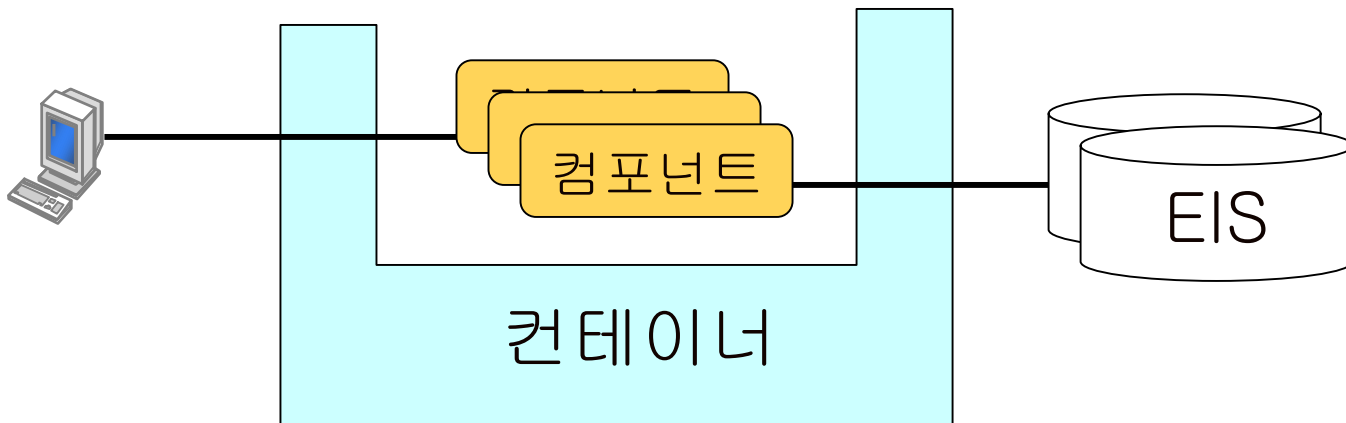
컨테이너의 관리/도움 하에서 작업을 수행하는 사용자 모듈

□ 컨테이너?

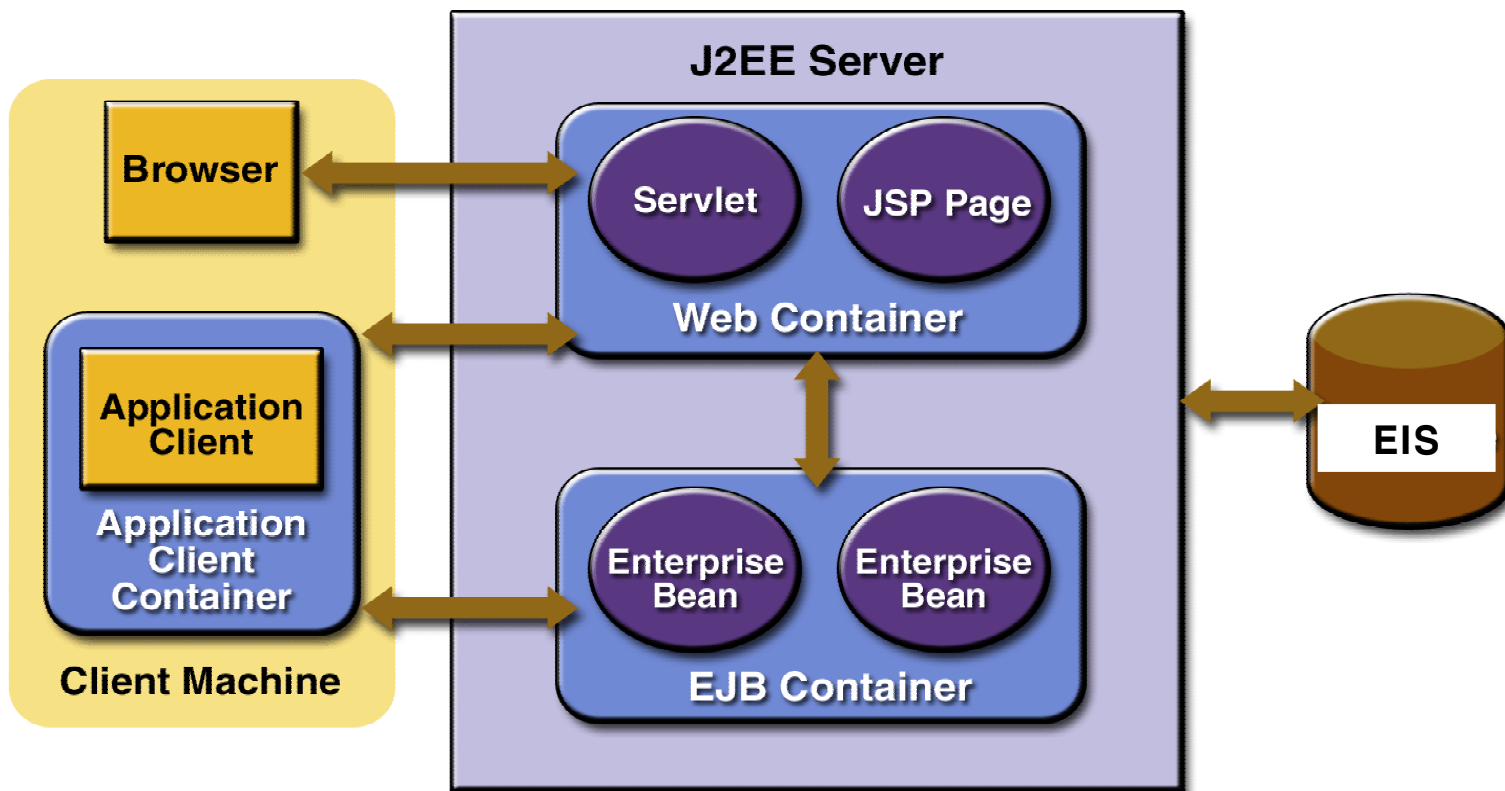
컴포넌트의 동작을 관리하고 도와주는 엔진(모듈들의 집합)

□ 모듈?

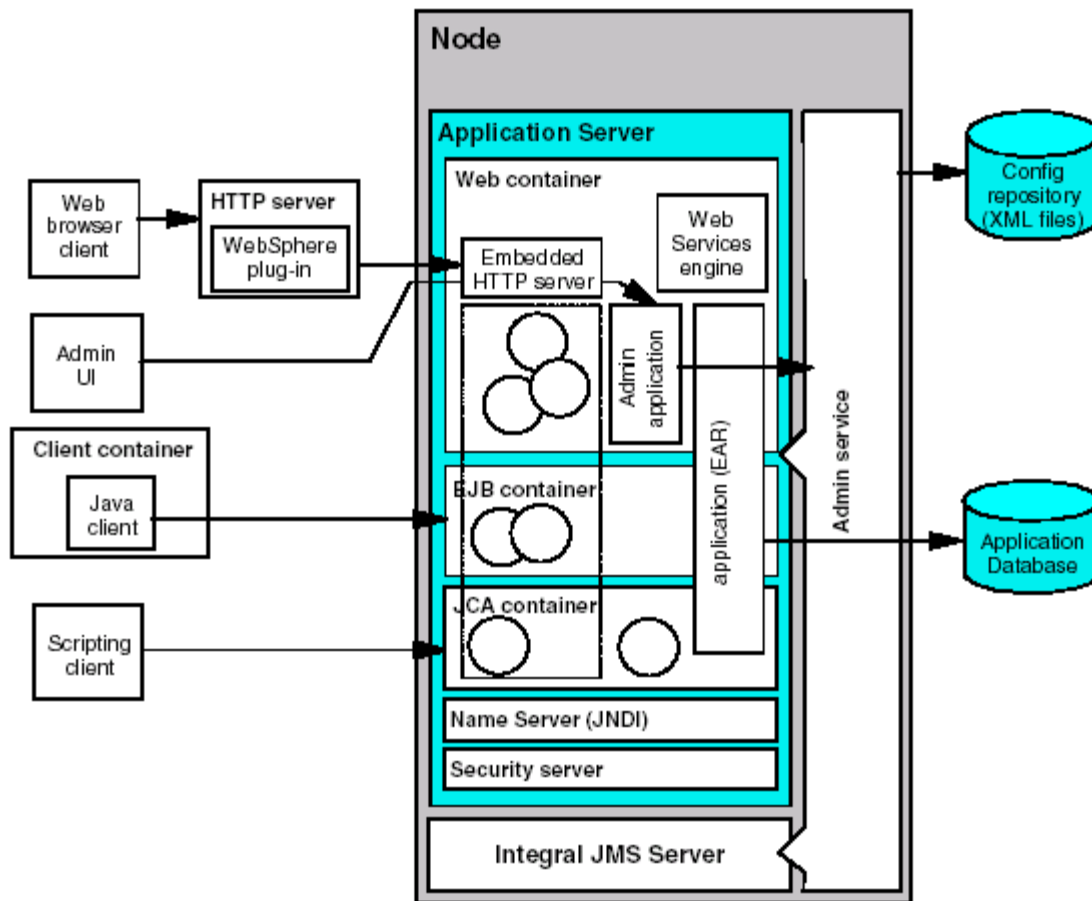
클래스/인터페이스 - in Java



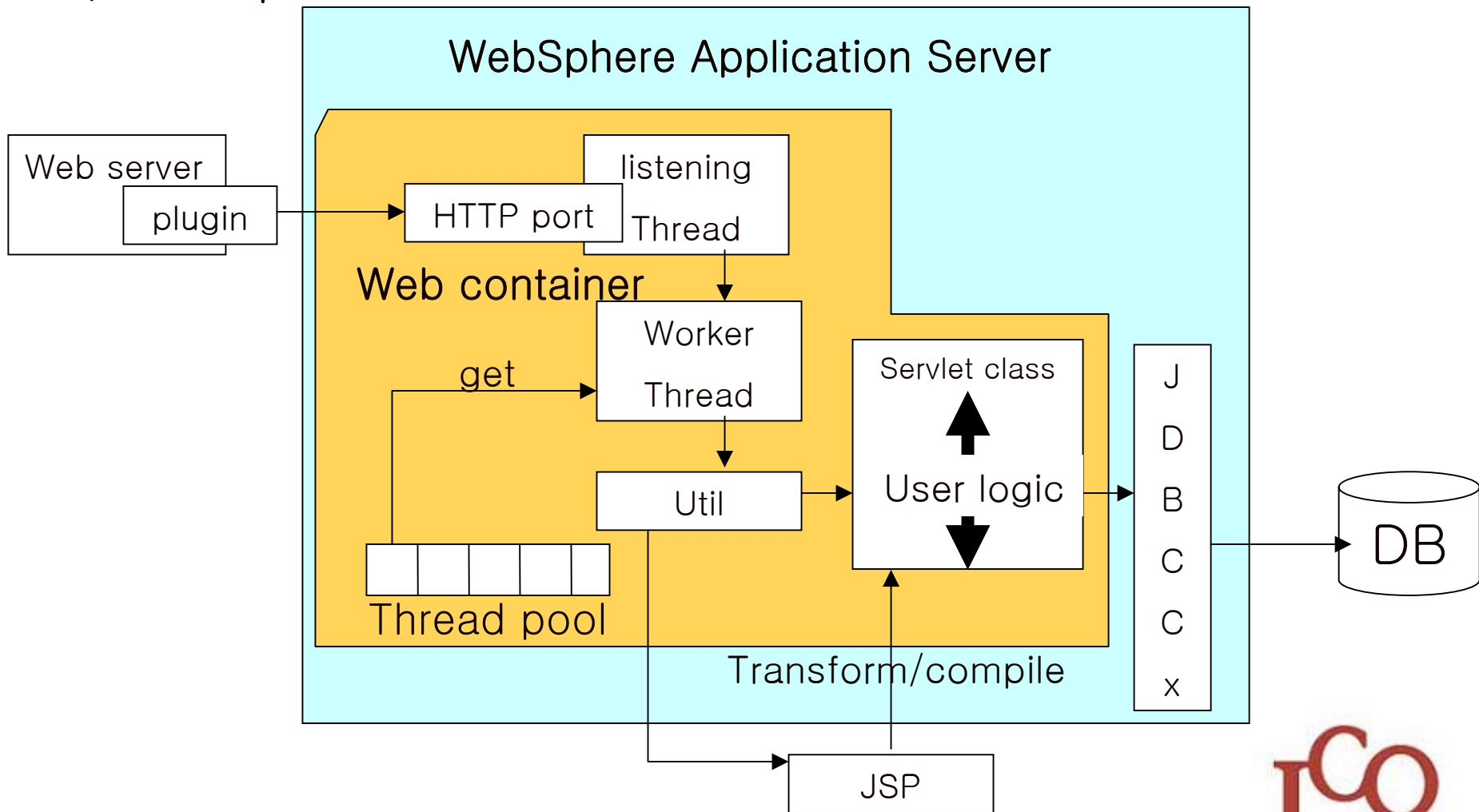
J2EE 아키텍처



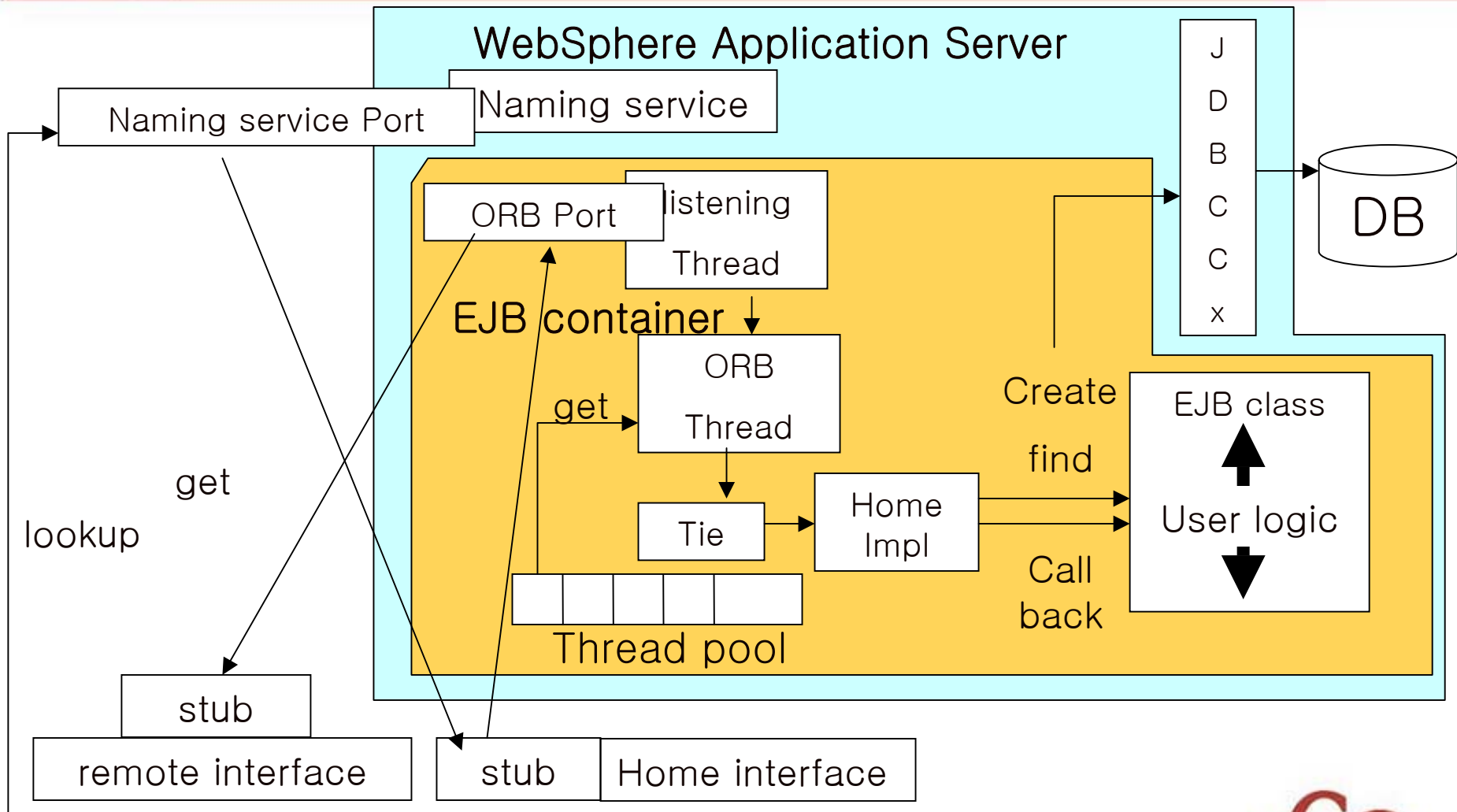
Ex) WebSphere V5



Ex) WebSphere V5

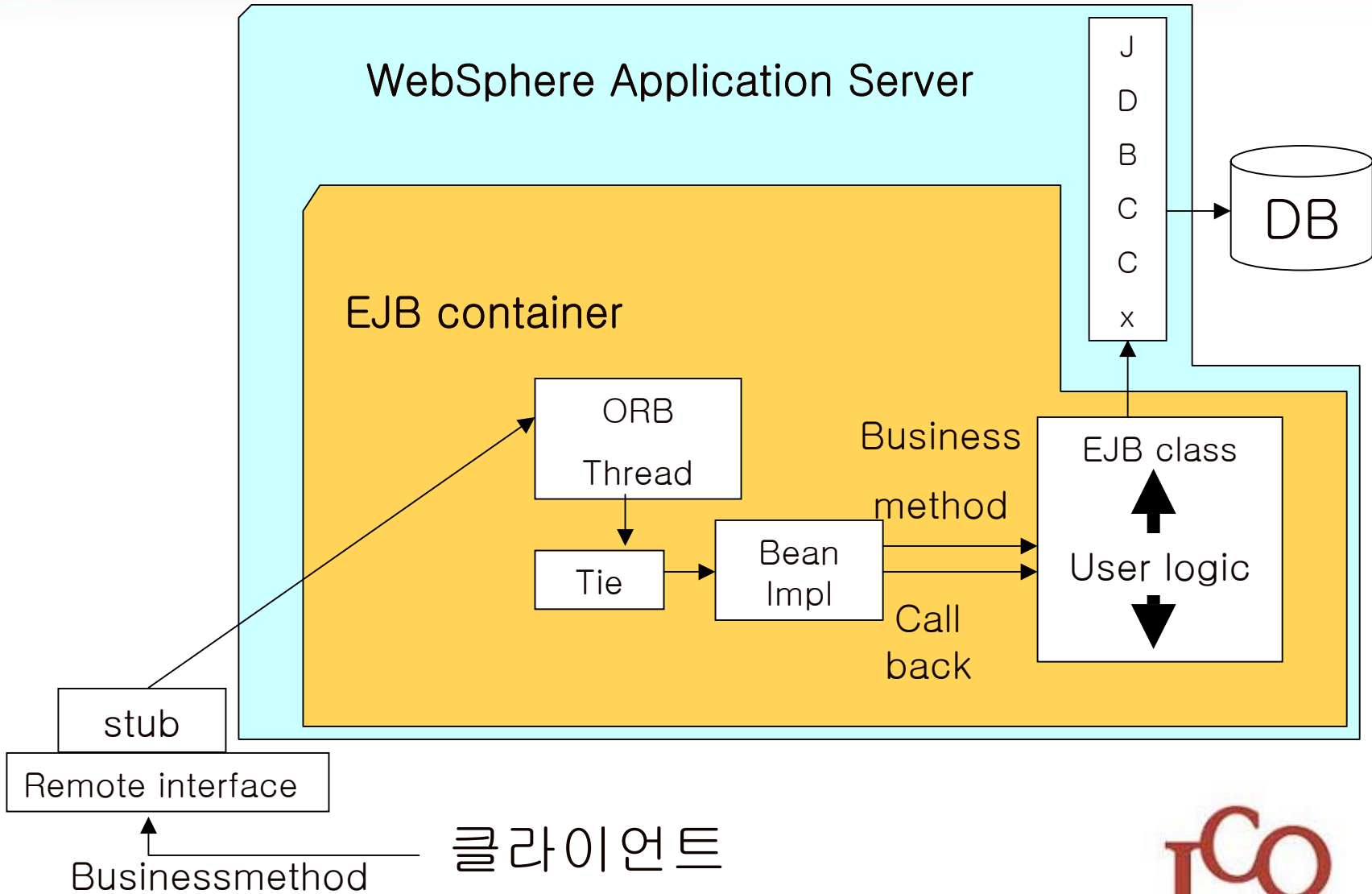


EJB 요청의 흐름 (Remote home call)

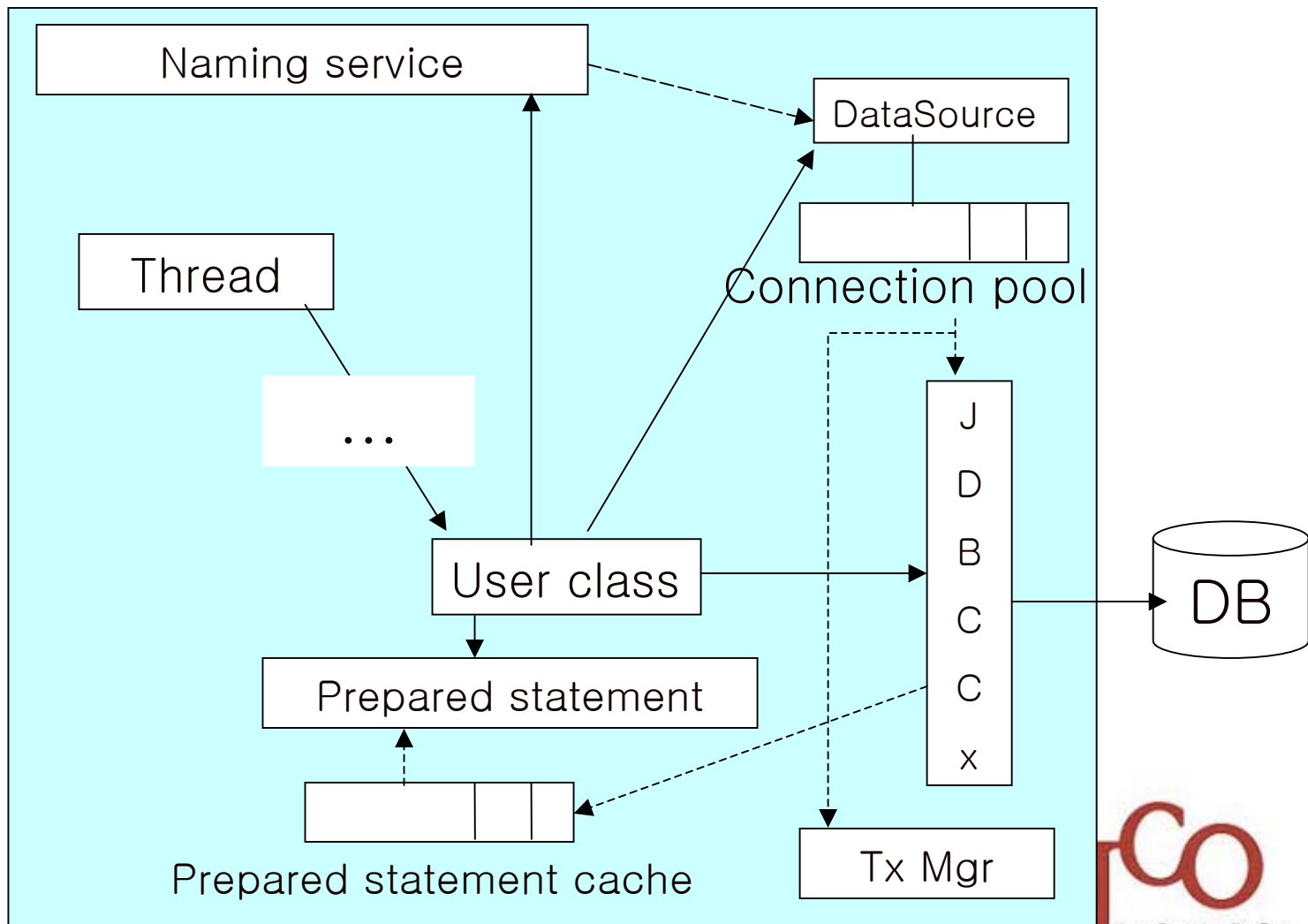


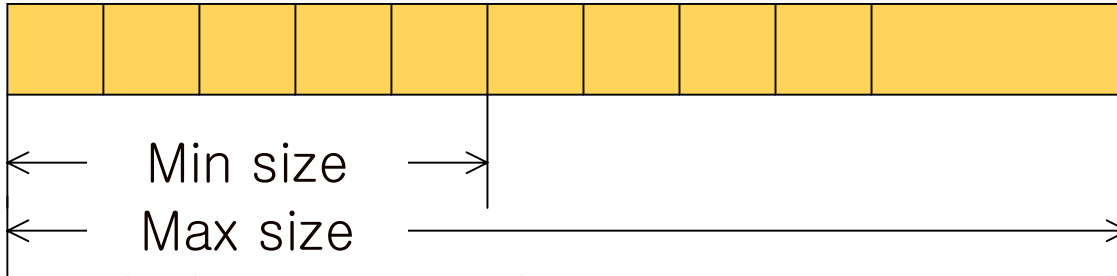
클라이언트

EJB 요청의 흐름 (Remote EJB Call)



Ex) WebSphere V5





- EJB Instance pool

- Servlet thread pool, ORB thread pool

- Other parameters -Ex) WebSphere V5

- 스레드 비활동 제한 시간

- 최대 크기를 초과한 할당 허용 여부

- JDBC Connection pool

- other parameters -Ex) WebSphere V5

- 연결 제한 시간 -Reap 시간

- 유효 제한 시간 -미사용 제한 시간

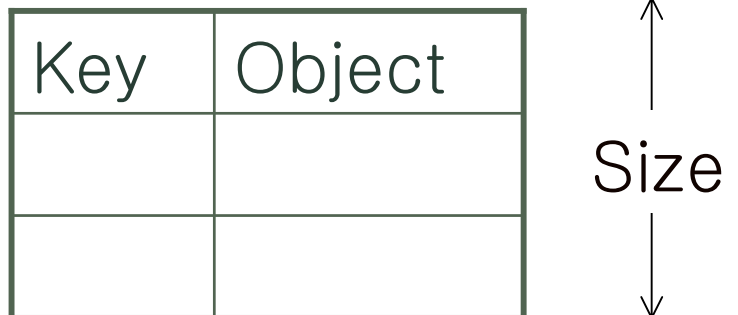
- 제거 정책

Ex) WebSphere V5

- Connection caching
- Prepared Statement caching

- Data caching
 - other parameters - Ex) WebSphere
 - 기본 우선 순위
 - 디스크 오프로드

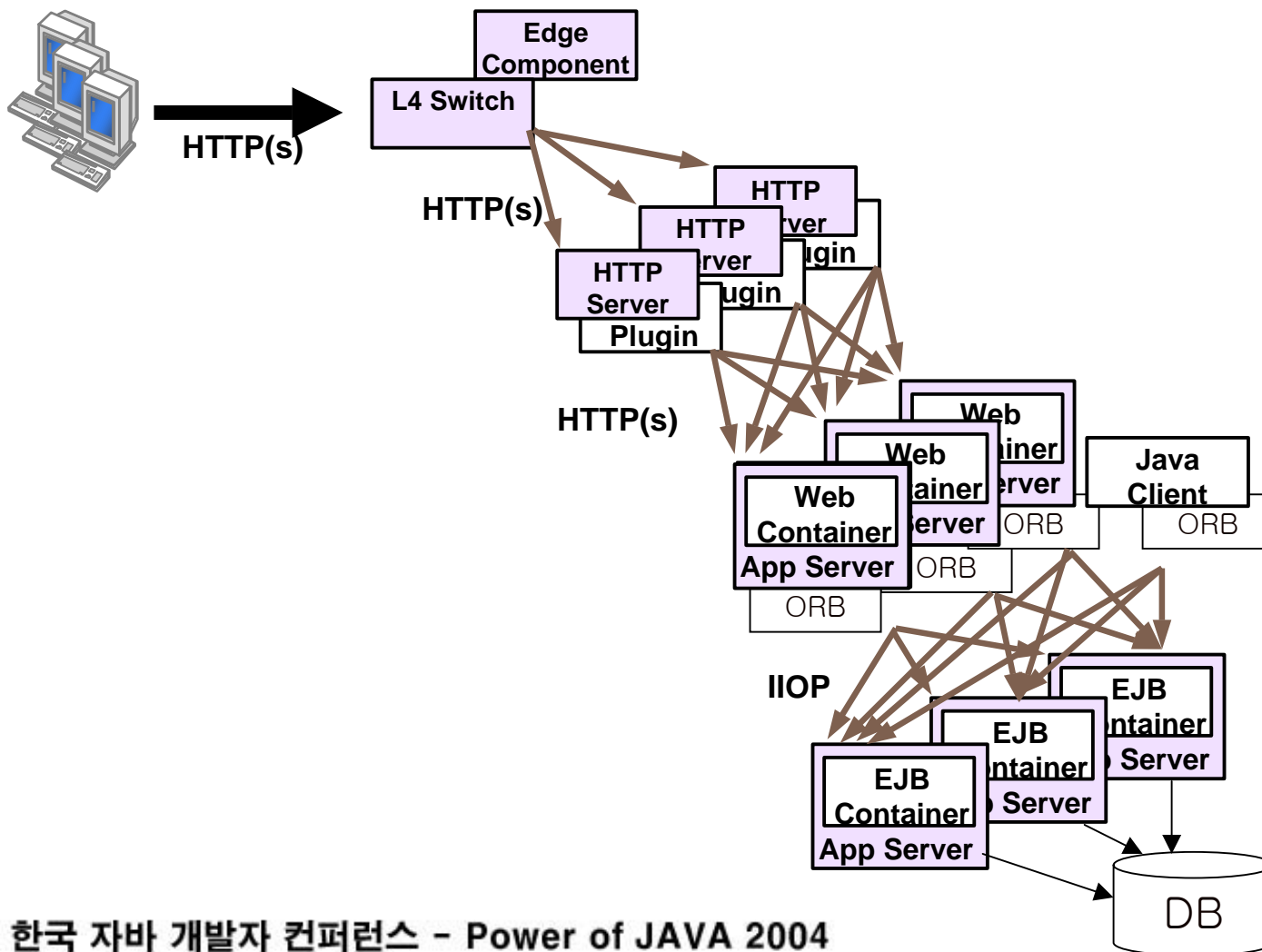
Hashtable



- Global Transaction, Local Transaction
- Transaction aware object의 종류
 - EJB, JDBC Connection... : Transaction aware
 - Etc : Transaction not aware
- XADataSource, Non-XA DataSource
- Resource Locking
- EJB Transaction Spec
 - EJB Transaction Attribute
 - Transaction Scope
 - Transaction Rollback condition
 - Transaction Isolation Level

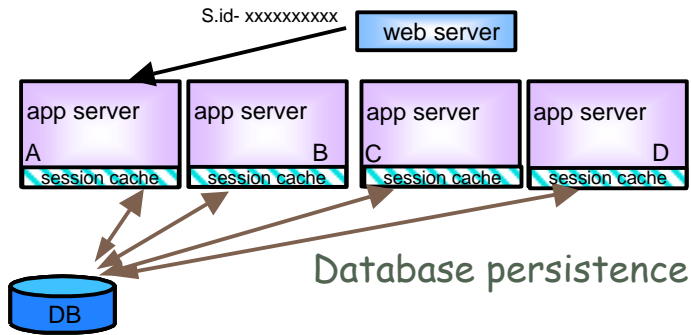
Clustering

Ex) WebSphere V5



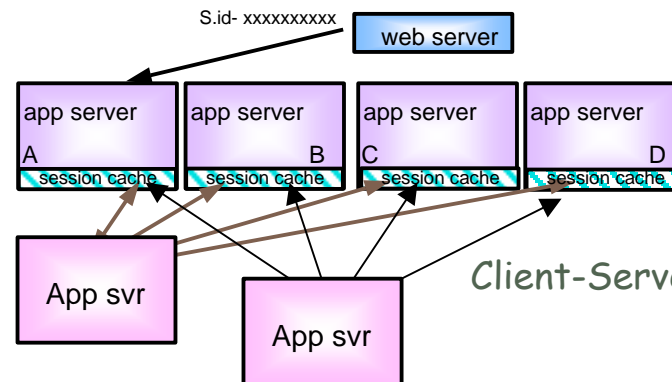
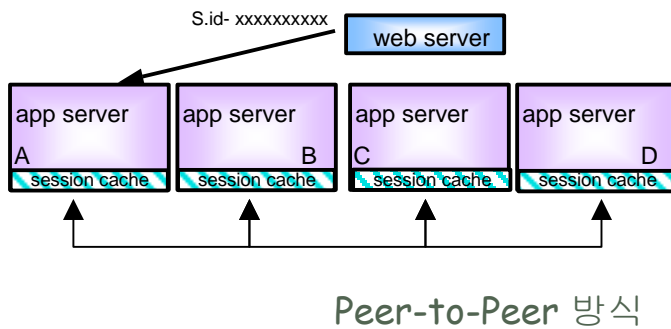
Http Session clustering

Ex) WebSphere V5



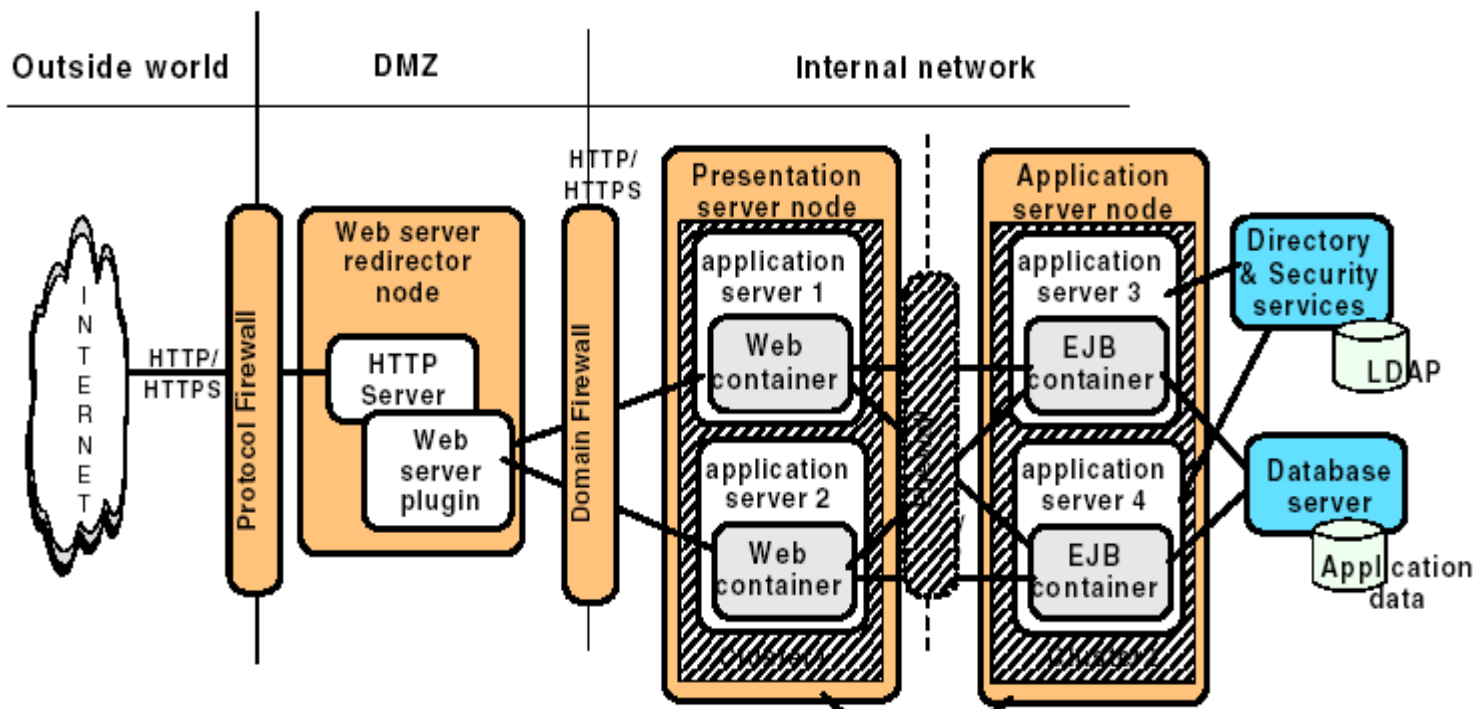
database persistence

Memory to Memory Replication



Client-Server 방식

Firewall



J2EE App = Java App

- J2EE 컨테이너, 컴포넌트, J2EE 어플리케이션은 JVM위에서 돌아가는 하나의 자바 어플리케이션일 뿐이다.

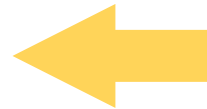
∴ JVM, JRE, JDK에 대한 이해는 J2EE 컨테이너의 동작을 이해하는 데 있어서 필수

-JVM의 확장 옵션, GC 알고리즘 등 몇 가지 동작은 OS별, 버전별로 다르다.

J2EE App = Java App

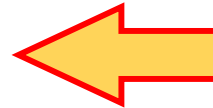
□ JVM이 하는 일

- Binary Code Analysis
- Heap, Stack 운영
- Garbage Collection



JVM에 의존적
(JVM의 기능을 그대로 이용)

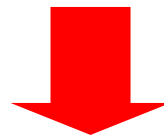
- Binary Code Loading
- Object Loading, Linking



상당부분 컨테이너에 의존적
(ClassLoader Customizing)

JVM and J2EE 컨테이너

- 다음의 조작을 통해 JVM의 런타임 동작에 영향을 줄 수 있다.
 - JVM에 대한 런타임 파라미터 추가/변경
 - JVM/JDK의 버전 변경
 - JDK의 라이브러리 변경 (디컴파일->변경->재컴파일)및 패치



□JVM 런타임 파라미터의 추가/변경 방법

□컨테이너에 종속적

□툴을 이용하는 방법

부트 클래스 경로	/usr/local/euijung/cha.jar:/usr/local/euijung/jco
상세한 클래스 로딩	<input type="checkbox"/>
상세한 가비지 콜렉션	<input checked="" type="checkbox"/>
상세한 JNI	<input type="checkbox"/>
초기 힙 크기	0
최대 힙 크기	256

□환경 변수나 서버 구동용 쉘 파일을 직접 수정

...

```
@REM set DEBUG=-Djava.compiler=NONE -Xdebug -Xnoagent -Xrunjdpw:transport=dt_socket,server=y,suspend=y,address=7777
"%JAVA_HOME%\bin\java" %WAS_TRACE% %DEBUG% %CONSOLE_ENCODING% "%CLIENTSAS%" "-
classpath" "%WAS_CLASSPATH%" "-Dws.ext.dirs=%WAS_EXT_DIRS%" %USER_INSTALL_PROP% "-
Dwas.install.root=%WAS_HOME%" "com.ibm.ws.bootstrap.WSLauncher" "com.ibm.ws.management.tools.WsServerLauncher" "%CONFIG_RO
OT%" "%WAS_CELL%" "%WAS_NODE%" %*
```

...

- TCP/IP 이론
- OS Memory, virtual memory 운용
- UNIX Shell, Shell Script, DOS Script