

**웹서비스와 서비스 지향**

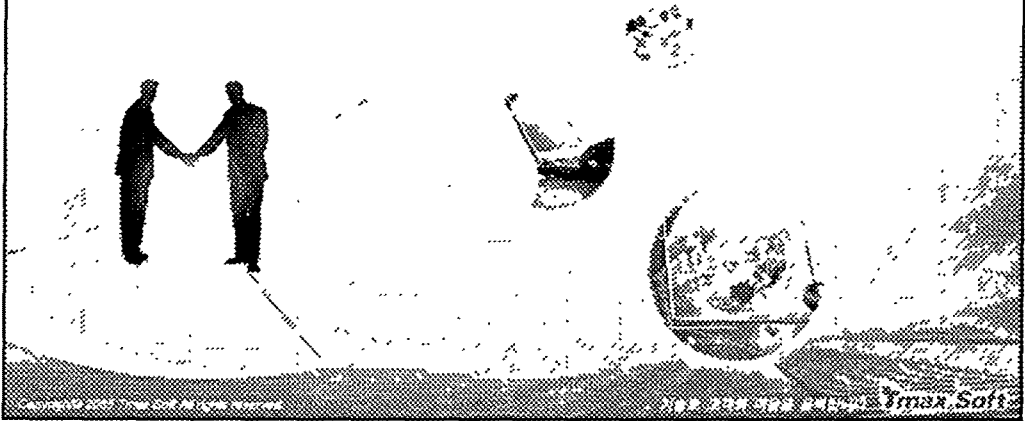
**아키텍처 구현**

***김택종(티맥소프트)***



# 웹서비스와 서비스 지향 아키텍처(SOA) 구현 - TmaxSoft -

2005. 10.27



## Contents

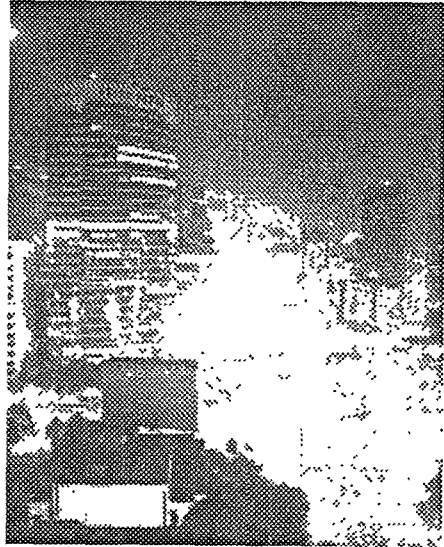
- I 회사 개요
- II 웹서비스의 구현
- III SOA 프레임워크

# 1. 회사소개

## 회사 개요

- 설립 : 1997년 6월 3일  
대표이사 : 김병국  
자본금 : 30억원  
매출 : 253억원 (FY2004)  
직원수 : 520명 (2005년 8월말 현재)
- 해외법인·지사 20명
  - 연구원 기술인력 4160여명
- 위치
- 본사 : 삼성역 글래스타워 18층
  - Tmax R&D센터 : 분당 서현역
  - 해외 3개 법인 (美 中 日)

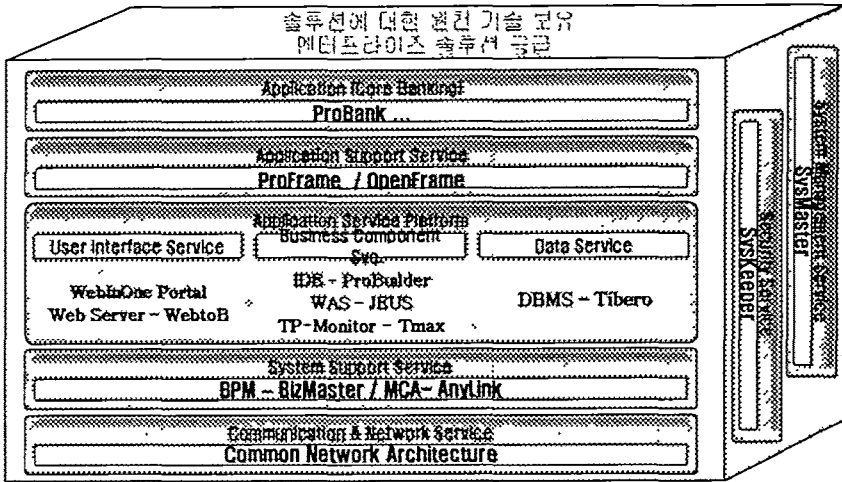
"티맥스가 있는 한  
글래스타워 18층의 불빛은  
꺼지지 않을 것입니다."



Tmax Soft

## 제공 Solution

Tmax Soft는 인터넷 시스템에서부터 기간제 시스템까지의 모든 시스템을 구축하기 위한 인프라 솔루션을 자체 기술로 제공하고 있습니다.



## 토털 소프트웨어 솔루션

<b>Tmax</b> TP-Monitor	시스템의 과부하를 제어함으로써 시스템의 다운을 방지하고 처리속도 문제를 지점한 비용으로 해결할 수 있는 세계 제1의 트랜잭션 모니터링 솔루션
<b>JEUS</b> Application Server	기업이 원하는 다양한 웹 자바 프로그램을 쉽고 빠르게 구현할 수 있는 세계 최초 J2EE 1.4 국제 표준 인증의 애플리케이션 인프라 솔루션
<b>WebToB</b> Web Server	기업의 웹을 통한 대고객, 직원서비스를 위한 고도의 안정성과 빠른 부하처리 능력을 지닌 고성능 웹 서버
<b>Tibero</b> DBMS	기존의 데이터베이스와는 다른 설계로 성능 및 안정성을 획기적으로 향상시킴으로써 기업의 핵심 데이터를 관리하는 차세대 데이터베이스 솔루션
<b>BizMaster</b> BPM/EAI	기업의 업무 프로세스를 종합적으로 관리하고 자동화함으로써 기업의 변화 대응시간 및 비용을 대폭 축소할 수 있는 비즈니스 프로세스 관리 솔루션
<b>ProFrame</b> Application Framework	차세대 업무시스템 구축시 애플리케이션의 확장성과 변화 대응능력을 높임으로써 유지보수 비용을 획기적으로 절감함과 동시에 시장 대응력을 향상시킬 수 있는 신개념의 애플리케이션 프레임워크 솔루션
<b>OpenFrame</b> Mainframe Rehosting	기존 메인프레임 기반 프로그램과 자료를 빠르게 저렴하게 유닉스/리눅스 기반의 오픈환경으로 전환할 수 있는 고효율 리호스팅 솔루션

Tmax Soft

## 포털 소프트웨어 솔루션

<b>AnyLink</b> Multi-Channel Integration	기업 내외 복잡한 이기종 시스템의 연동을 단순화시킴으로써 IT 효율성을 대폭 향상시킬 수 있는 다중 채널 통합 솔루션
<b>SysMaster</b> APM	기업 내 애플리케이션 및 시스템들을 종합적으로 모니터링하고 분석함으로써 전산 인력의 생산성을 향상시킬 수 있는 성능 관리 솔루션
<b>ProBuilder</b> IDE	클라이언트/서버 개발환경뿐만 아니라 웹서비스 개발환경까지 완벽하게 지원하는 편리한 개발자용 개발 도구
<b>SysKeeper EAM</b> EAM	기업 내 e비즈니스 상에서 발생하는 다양한 보안 문제를 해결할 수 있는 전자적 인증·권한 관리 솔루션
<b>SysKeeper OS</b> Secure OS	각종 해킹으로부터 서버의 데이터와 애플리케이션을 원천적으로 보호할 수 있는 서버보안 솔루션
<b>SkyMail</b> Mail Server	기존 메일서버의 보안 문제를 개선하면서 안정성과 성능을 대폭 향상시킨 웹메일 솔루션
<b>WebInOne Portal</b> Portal	기업 내외부의 다양한 정보와 업무 프로세스를 단일 화면을 통해 편리하게 제공함으로써 직원, 고객, 파트너사의 생산성을 극대화할 수 있는 기업 포털 솔루션

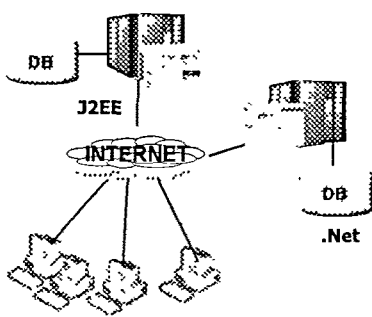
Max Soft

## II. 웹서비스의 구현

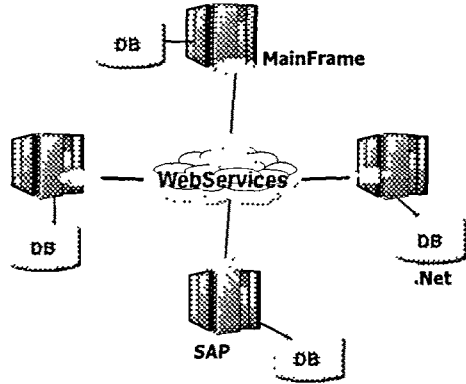
## 웹서비스 정의

### 웹서비스(Web Services)

- 플랫폼 및 기반 언어가 서로 다른 이기종 시스템 간의 서비스 호출 및 제공을 가능하게 하는 기술
- MS, IBM의 강력한 지원에 힘입어 시스템 통합 표준으로 각광받고 있음



기존 웹 시스템



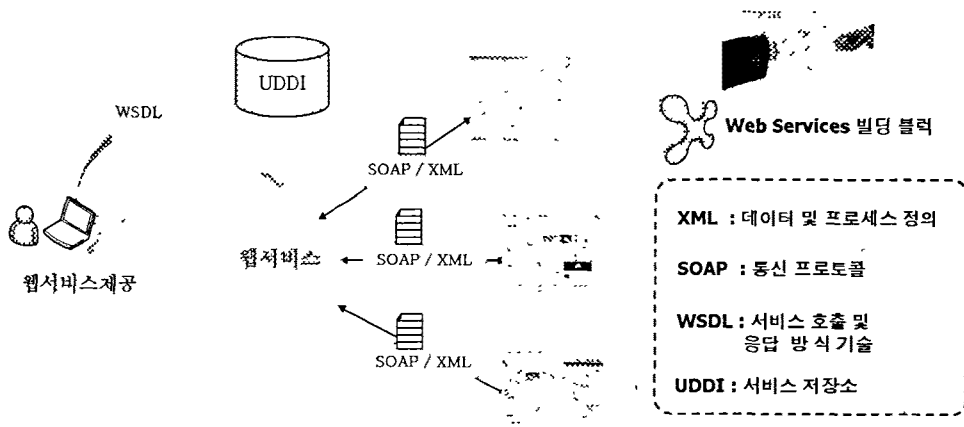
웹 서비스 기반 시스템

Emax Soft

## 웹서비스 아키텍처

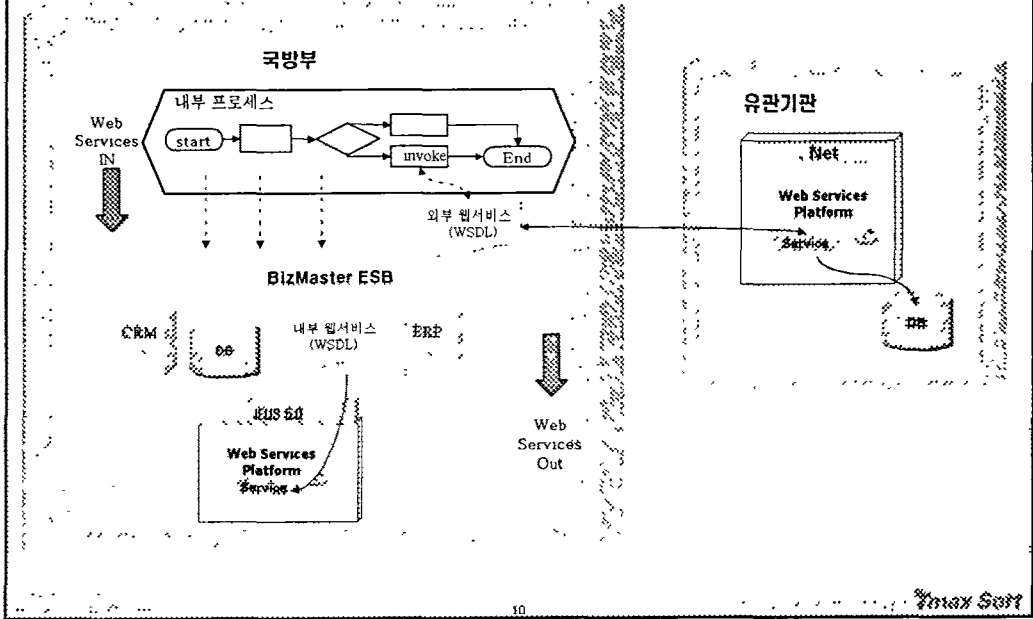
### 웹서비스(Web Services) 아키텍처

- 웹서비스는 서비스 create, publish, search, bind 네 단계로 구성됩니다.

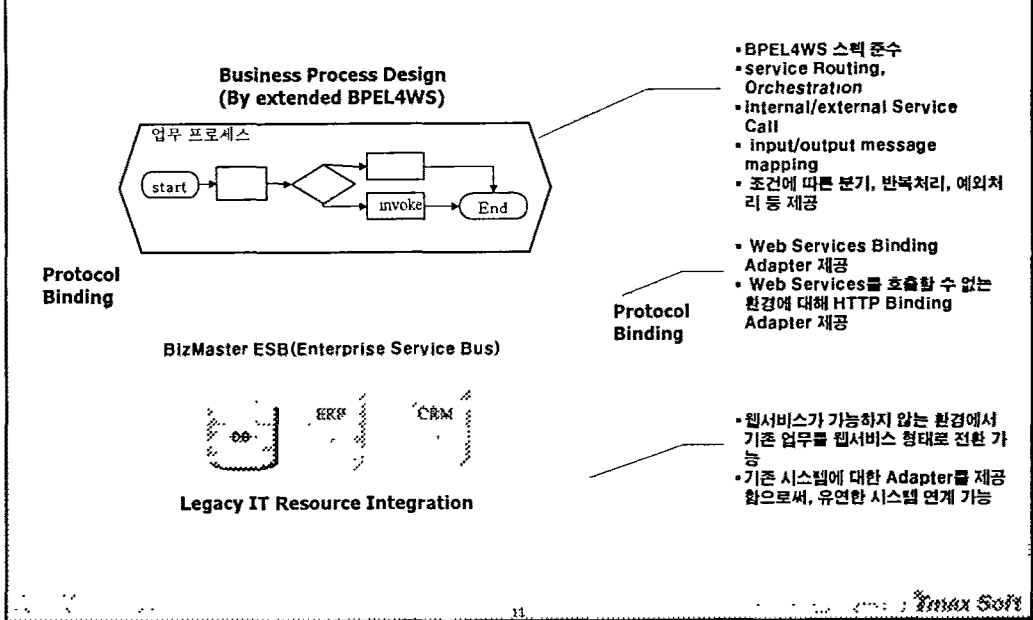


Emax Soft

# Tmax 웹서비스 개요

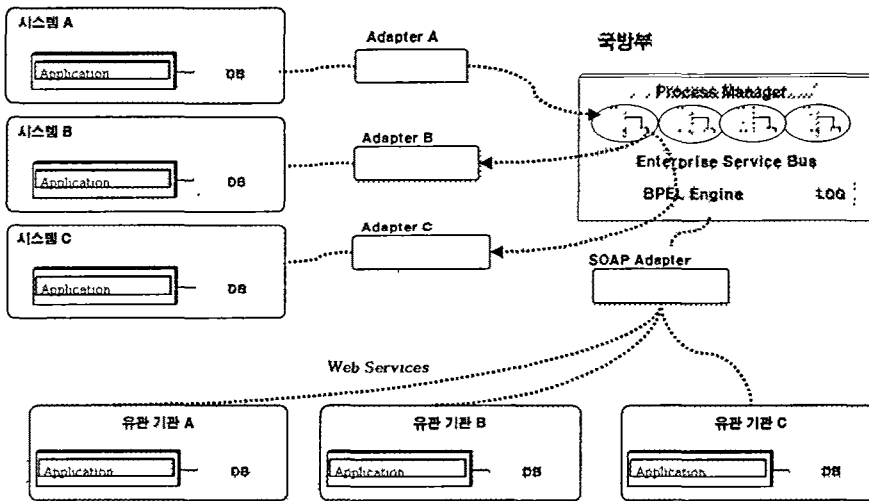


# BizMaster ESB의 역할

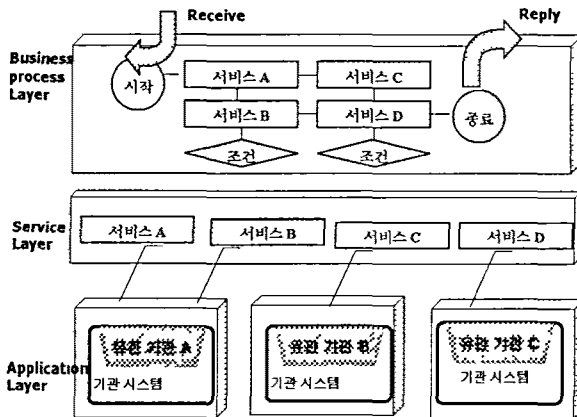




# ESB - Application Integration



# ESB - 협업적 웹서비스 제공



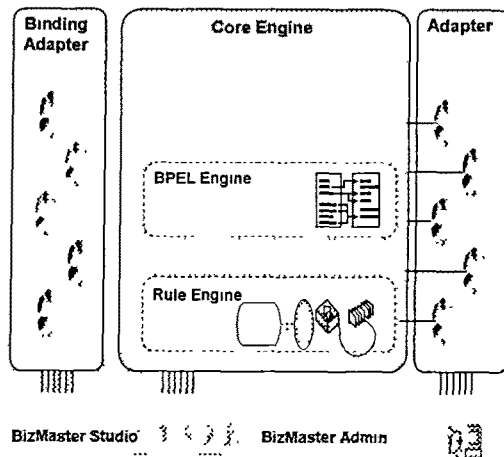
## 협업적 웹서비스 도입효과

- 향후 웹서비스 확대 시, 협업적 웹서비스 기반 마련
- 프로세스 자동화를 통한 업무 단계 단축, 업무 효율성 향상
- 프로세스의 유연한 변경, 집중화된 프로세스 관리
- 기존 서비스를 이용하여 새로운 비즈니스 가치 생산

- 협업적 웹서비스는 기존 웹서비스들을 이용해 하나의 자동화된 비즈니스 프로세스를 정의하는 것을 말함
- 유관 기관들의 웹서비스들을 서로 조합하여 국방부의 새로운 협업 프로세스 모델을 정의함

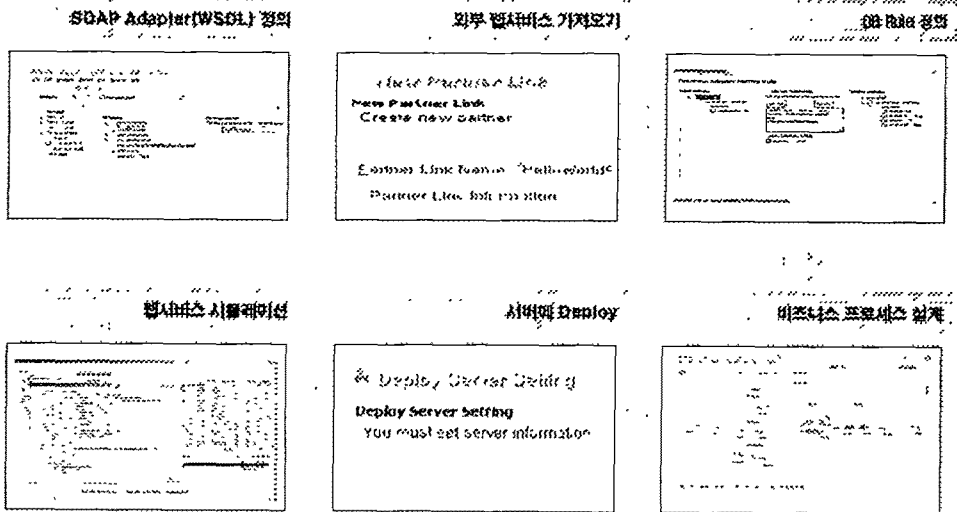
정보 공동활용에서 업무 협업으로 시스템 발전

# BizMaster 제품 아키텍처



컴포넌트	설명
BPEL 엔진	<ul style="list-style-type: none"> <li>WSBPEL 자동 실행 엔진</li> <li>동기 및 비동기 웹서비스를 모두 지원</li> <li>XML 쿼리의 언어인 XPath 표현식 처리 및 XPath 함수 처리가능</li> <li>XML Schema 처리 및 XML Schema에 따른 XML 인스턴스 문서 생성</li> <li>WSDL 정의에 따른 메시지 처리 가능</li> </ul>
Rule 엔진	<ul style="list-style-type: none"> <li>비즈니스 Rule을 자동으로 검색하여 실행하는 엔진</li> <li>비 프로그램 방식으로 비즈니스 Rule 시스템 개발</li> </ul>
리소스 어댑터	<ul style="list-style-type: none"> <li>다양한 전산 자원이거나 애플리케이션과 비즈마스터와의 연계 및 통신을 자동화 해주는 컴포넌트로, 시스템 통합의 핵심 요소</li> <li>비즈마스터는 파일 어댑터, 데이터베이스 어댑터, SAP 어댑터 등 다양한 리소스 어댑터를 제공하며 또, 필요에 따른 다양한 어댑터 계속 추가</li> </ul>
프로토폴 바인딩 어댑터	<ul style="list-style-type: none"> <li>내부의 정규화된 메시지를 전송 프로토콜에 따라 적합한 메시지로 변환하여 통신을 가능하게 해주는 프로토폴 처리기</li> <li>외부로부터의 호출을 받아주는 서버 모드와 외부의 서버를 호출하는 클라이언트 모드를 지원</li> </ul>
BizMaster Studio	<ul style="list-style-type: none"> <li>효율적인 WSBPEL 모델링을 위한 프로세스 설계 툴</li> </ul>
BizMaster Admin	<ul style="list-style-type: none"> <li>BizMaster의 모든 컴포넌트를 관리할 수 있는 그래픽 사용자 인터페이스 도구</li> </ul>

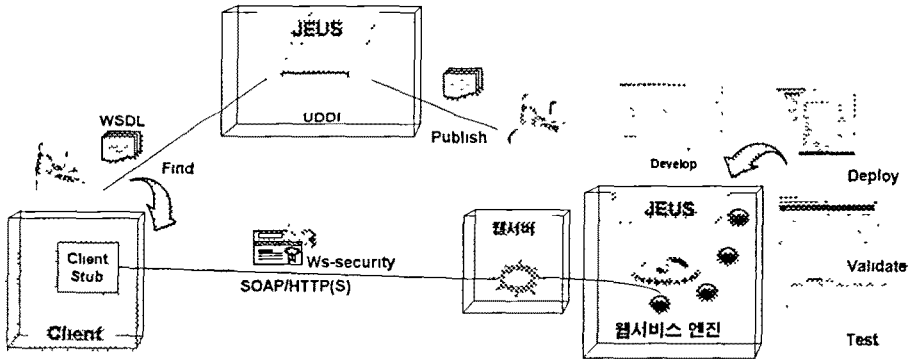
# BizMaster 웹서비스 작성 과정



## JEUS 5.0 WAS의 웹서비스 개요

### JEUS 웹서비스

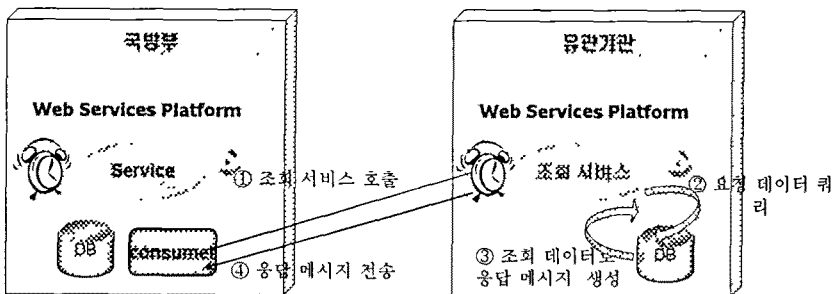
- JEUS 5.0은 웹서비스 기본 스펙인 SOAP 1.2, WSDL 1.1, UDDI 3.0과 보안, 신뢰성 메시징, 단기성 트랜잭션 보장, 성능 Matrix 모니터링 등 지원
- JMaker를 통해 개발, 배치, 모니터링, 테스트 전 과정을 손쉽게 수행 가능



16

Tmax Soft

## JEUS 5.0 WAS의 웹서비스 구현 모델



- ◆ one-way 모델로 구축 시, ① 조회 서비스 호출 후, 바로 리턴, 배치 처리 가능
- ◆ 서비스 요청 프로그램을 스케줄러에 등록하여 특정 시간의 배치 처리 가능

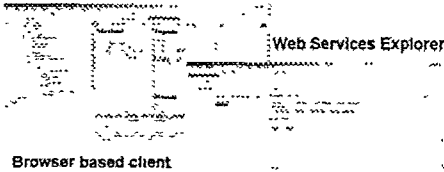
17

Tmax Soft

# JMaker를 통한 편리한 웹서비스 개발환경 제공

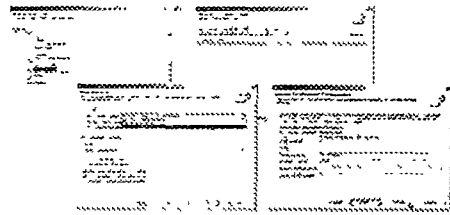
## 1. 다양한 개발 방식 지원

- TOP-DOWN 개발 방식 지원
  - WSI 먼저 작성, WSDL을 토대로 서비스 인터페이스 자동 생성
- Bottom-Up 개발 방식 지원
  - 서비스 인터페이스 먼저 작성, 서비스 인터페이스를 토대로 WSDL 자동 생성



## 2. 테스트 용의한 환경 제공

- 브라우저 기반의 테스트 클라이언트 자동 생성 후 테스트 실시
- Web Services Explorer를 통해 테스트 실시



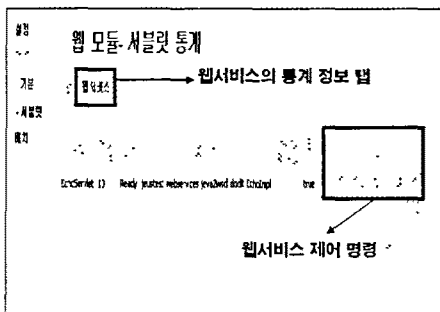
## 3. 손쉬운 XML 작성 및 편집

- XML Code Completion 기능 제공
- WSDL, XML Schema GUI Editor 제공
- XML Code Validation 기능 제공

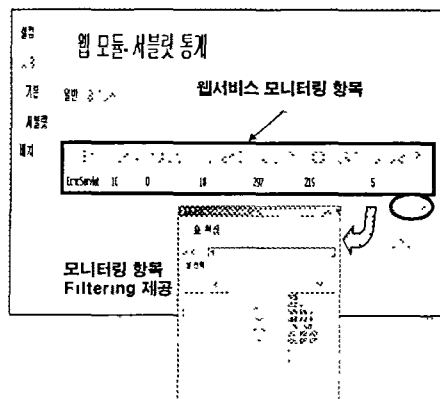


# GUI 관리 툴을 통한 웹서비스 관리 용의성 제공

## 서비스 제어

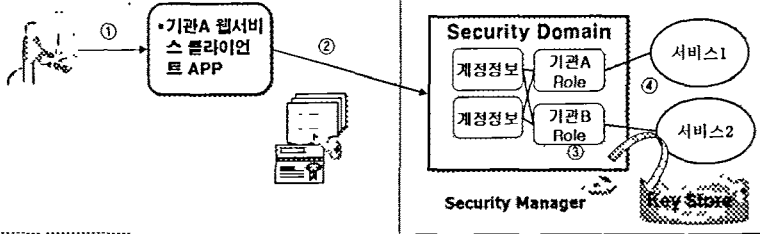


## 서비스 모니터링

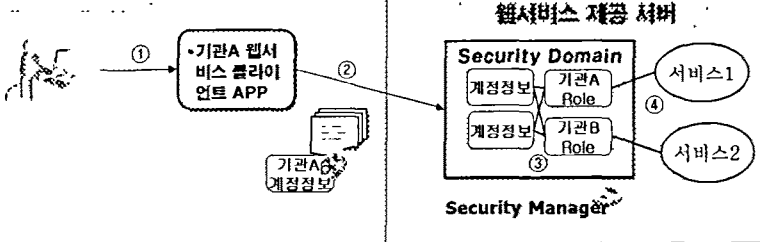


# 웹서비스 보안 및 사용자 접근 권한 관리

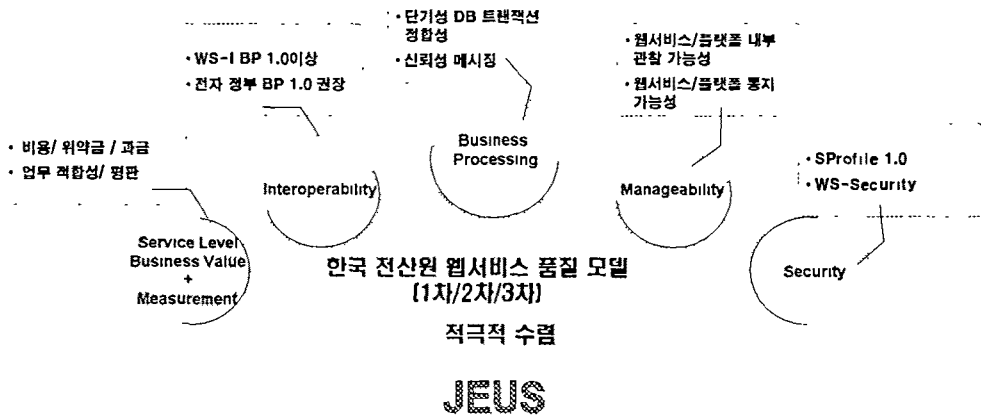
데이터 보안이 필요할 경우



데이터 보안이 필요 없을 경우



# JEUS 5.0 웹서비스의 특징



## JEUS

- JEUS 5.0 최초 웹서비스 품질 인증 시험 대상 (올 11월 예정)
- 향후 웹서비스 프로젝트 구축 시, 웹서비스 품질 요건 준수 여부가 플랫폼 선정 및 검수 시 주요한 기준으로 채택될 가능성이 높음
- 국내 원천 기술 보유로 제품 설계 단계에서부터 품질 요소 적극적 반영

## WAS별 웹서비스 지원 비교

지원 기능	JEUS 5.0	WebLogic 9.0	Oracle AS 10g	WebSphere 6.0
기본 기능	<ul style="list-style-type: none"> <li>• SOAP 1.2, WSDL 1.1, JAX-RPC 1.1, SAAJ 1.2</li> <li>• Built-In UDDI Server 제공, UDDI 2.0/3.0(부분), JAX-R 1.0</li> </ul>	<ul style="list-style-type: none"> <li>• SOAP 1.2, WSDL 1.1, JAX-RPC 1.1, SAAJ 1.2</li> <li>• Built-In UDDI 2.0 Server 제공, UDDI 2.0, JAX-R 1.0</li> </ul>	<ul style="list-style-type: none"> <li>• SOAP 1.2, WSDL 1.1, JAX-RPC 1.1, SAAJ 1.2</li> <li>• Built-In UDDI 2.0 Server 제공, UDDI 2.0, JAX-R 1.0</li> </ul>	<ul style="list-style-type: none"> <li>• SOAP 1.2, WSDL 1.1, JAX-RPC 1.1, SAAJ 1.2</li> <li>• IBM Private UDDI Server 제공, UDDI 2.0, JAX-R 1.0</li> </ul>
신뢰성 메시지	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (WS-RM 1.0)</li> </ul>	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (WS-RM 1.0)</li> </ul>	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (But, WS-Reliability)</li> </ul>	<ul style="list-style-type: none"> <li>• No</li> </ul>
보안	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (WS-Security 1.0)</li> </ul>	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (WS-Security 1.0)</li> </ul>	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (WS-Security 1.0)</li> </ul>	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (WS-Security 1.0)</li> </ul>
상호 운용성	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (WS-I Basic Profile 1.0 Compliance, Net Interoperability)</li> </ul>	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (WS-I Basic Profile 1.0 Compliance, Net Interoperability)</li> </ul>	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (WS-I Basic Profile 1.0 Compliance, Net Interoperability)</li> </ul>	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (WS-I Basic Profile 1.0 Compliance, Net Interoperability)</li> </ul>
트랜잭션	<ul style="list-style-type: none"> <li>• Yes</li> <li>• WS-Atomic Transactions, WS-Coordination</li> </ul>	<ul style="list-style-type: none"> <li>• No</li> </ul>	<ul style="list-style-type: none"> <li>• No</li> </ul>	<ul style="list-style-type: none"> <li>• Yes</li> <li>• WS-Atomic Transactions, WS-Coordination</li> </ul>
비동기 메시징	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (Call Back methods, WS-Addressing을 이용한 비동기 호출 지원)</li> </ul>	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (Asynchronous request-response, Buffered web services, JMS Transport)</li> </ul>	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (JMS Web Services)</li> </ul>	<ul style="list-style-type: none"> <li>• Yes</li> <li>• (SOAP over JMS)</li> </ul>
기타 부가 기능	<ul style="list-style-type: none"> <li>• MOWS(Management Of Web Services)의 Metrics 항목을 MBean으로 지원함                             <ul style="list-style-type: none"> <li>- MaxResponseTime</li> <li>- LastResponseTime</li> <li>- ServiceTime</li> <li>- NumberOfRequests</li> <li>- NumberOfSuccessfulRequests,</li> <li>- NumberOfFailedRequests</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Web Services Metadata for the Java Platform (JSR-181)에 의한 Web Services Annotation Programming Model 지원</li> </ul>	<ul style="list-style-type: none"> <li>• Extended service implementer 제공 :                             <ul style="list-style-type: none"> <li>• Stateless PL/SQL,</li> <li>• Stateful Java Class</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• WSIF (Web Services Invocation Framework)를 통해 service의 포맷이나 transport protocol에 Independent하게 웹서비스를 호출할 수 있는 환경 제공</li> </ul>

22

Imax Soft

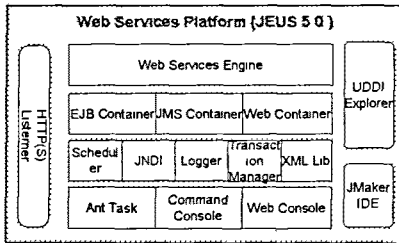
## 한국 전산원의 웹서비스 방향

	1단계 (2003)	2단계	3단계
적용 분야	<ul style="list-style-type: none"> <li>• 기업 내 A2A 통합</li> <li>• 초기 1:1 B2B 통합</li> </ul>	<ul style="list-style-type: none"> <li>• 기업간 정적 비즈니스 프로세스 통합</li> </ul>	<ul style="list-style-type: none"> <li>• 기업간 동적 비즈니스 프로세스 통합</li> </ul>
비즈니스 가치 품질	<ul style="list-style-type: none"> <li>• 비용/위약금</li> <li>• 적합성</li> </ul>	<ul style="list-style-type: none"> <li>• 평판</li> </ul>	<ul style="list-style-type: none"> <li>• 미터링/과금</li> </ul>
서비스 레벨 측정 품질	<ul style="list-style-type: none"> <li>• 성능 (비트랜잭션 서비스)</li> <li>• 안정성 (이용가능성, 신뢰성)</li> </ul>	<ul style="list-style-type: none"> <li>• 성능 (트랜잭션 서비스)</li> <li>• 안정성 (접근성)</li> </ul>	<ul style="list-style-type: none"> <li>• 성능 (협업적 서비스)</li> </ul>
관리 가능성 품질	<ul style="list-style-type: none"> <li>• WSDM 표준 부분 권장</li> <li>• 웹서비스/플랫폼 내부 관찰성</li> <li>• 웹서비스/플랫폼 내부 통지 가능성</li> <li>• 자체 관리자용 통합 관리</li> </ul>	<ul style="list-style-type: none"> <li>• WSDM 표준 준수</li> <li>• 웹서비스/플랫폼 제어 가능성</li> <li>• 외부 관리자용 통합 관리</li> </ul>	<ul style="list-style-type: none"> <li>• 통합 관리 센터를 위한 관리 가능성 표준</li> <li>• 통합 관리 센터를 통해 관리되는 웹서비스 플랫폼</li> </ul>
상호 운용성	<ul style="list-style-type: none"> <li>• 표준 준수성</li> <li>• 전자정부 BP 1.0 상호 운용성</li> </ul>	<ul style="list-style-type: none"> <li>• 전자정부 SP 1.0 상호 운용성</li> </ul>	<ul style="list-style-type: none"> <li>• 확장</li> </ul>
보안 품질	<ul style="list-style-type: none"> <li>• 보안 프로파일 0, 보안 프로파일 1(옵션)</li> </ul>	<ul style="list-style-type: none"> <li>• 보안 프로파일 2, 보안 프로파일 3(옵션)</li> </ul>	<ul style="list-style-type: none"> <li>• 보안 프로파일 4</li> </ul>
비즈니스 처리 품질	<ul style="list-style-type: none"> <li>• 신뢰성 메시지</li> <li>• 단기 트랜잭션</li> </ul>	<ul style="list-style-type: none"> <li>• 장기 트랜잭션</li> <li>• BPEL의 정확한 사용</li> <li>• 중앙 집중 비즈니스 프로세스 관리</li> </ul>	<ul style="list-style-type: none"> <li>• 분산 비즈니스 프로세스 관리</li> </ul>

23

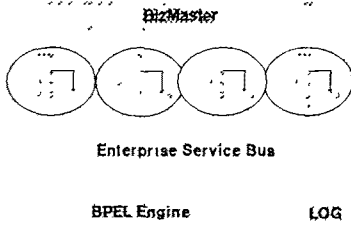
Imax Soft

# JEUS 5.0 / BizMaster 웹서비스 비교자료



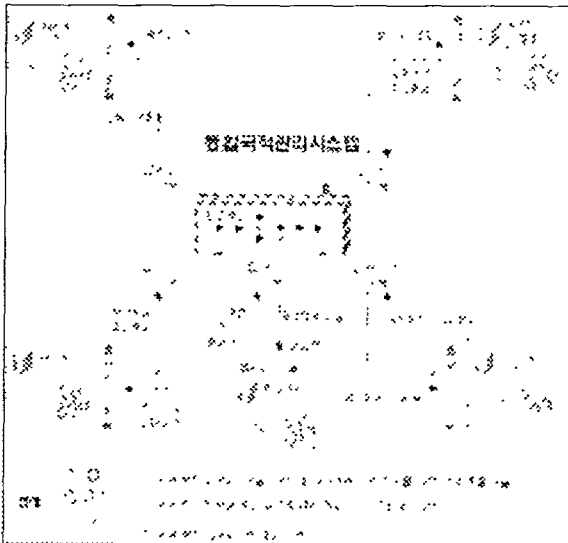
상호 보완적인  
제품

- JEUS 5.0 확장**
- Point to Point 메시징 보장
  - 웹서비스 플랫폼일 뿐 아니라, 인터넷 시스템 구축 인프라로 사용
  - 광범위한 도메인에서 사용 중인 미들웨어
  - J2EE 1.4 표준 스펙에 기반



- BizMaster 확장**
- Service Routing & composition의 Business Process Layer 제공
  - GUI 툴을 통한 프로세스 설계, 프로세스 변경 시 유연한 변경 가능, 복잡한 프로세스에 대한 중앙 집중적 관리 가능
  - 기존의 레거시 시스템에 대한 다양한 Adapter를 제공함으로써, 웹서비스를 제공하지 않는 환경에 대해 웹서비스로 전환 제공
  - JEUS 5.0 WAS 기반
  - 차세대 웹서비스 발전 방향과 부합되는 제품

# 협업적 웹서비스 구축 사례

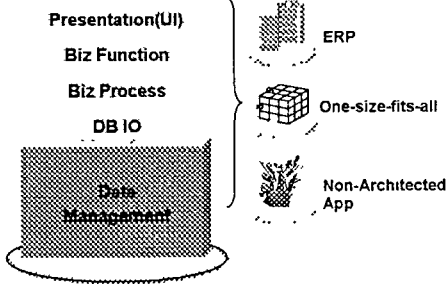


구분	상세내역
프로젝트 명	• 법무부 통합 국적 관리 시스템
개요	• 국적유관 기관과의 연계 체계를 웹서비스 기반으로 자동화하여, 통합국적관리시스템을 구축하는 프로젝트
구축 기간	• 2005.04 ~ 2005.11
특징	• 웹서비스를 통한 국적 유관 기관과의 연계 체계 구축 • BPEL(Business Process Execution Language)을 적용하여 웹서비스 협업 모델을 구축 함으로써, 협업 프로세스의 관리성 및 유지 보수성 향상
기대 효과	• 유관 기관과의 업무 연계를 자동화 함으로써, 국적 업무의 정확성과 효율성 제고 • 표준 웹서비스 기술 기반으로 확장성 확보, G4C 연계 기반 구축 • 프로세스 절차 간소화 및 민원 처리 상황 실시간 조희를 통한 대민 서비스 질 향상

# III. SOA 프레임워크

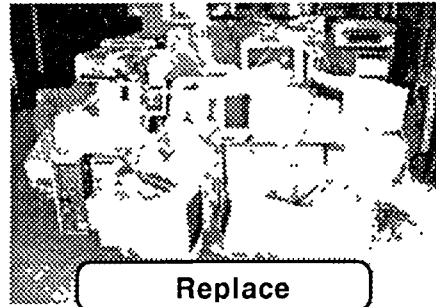
## Before SOA

### Monolithic Application



- 비즈니스 요구 변화를 고려하지 못함
- 각 비즈니스 부서의 요구사항을 공통화
- 패키지 도입의 경우 획일적 프로세스 강요
- 패키지 도입을 통한 경쟁 차별성은 무엇인가?

### IT Infrastructure



### Replace

- 인프라 교체에 따른 고비용 및 내재 위험성 상존
- 기존 인프라 유지보수에 따른 고비용 및 내재 위험성 상존
- 변화하는 비즈니스 지원을 위한 필수불가결한 운영 형태



## SOA기반의 프레임워크 도입 배경

- 비즈니스의 IT 의존도 증가 및 IT Enabler로서 '서비스' 대두
- Buzzword로서의 SOA 개념 명확화
- 분산 이기종의 개별 업무 시스템 (기간계/SCM/CRM/ERP 등)에 대한 유연한 통합/협업/변화/관리

비즈니스와 IT의 동기화  
(Alignment)



- IT서비스중심의 경영 전략 확산
- IT관련 Lifecycle이 Service Lifecycle관점으로 확장
- 'Plug-In' 형태로 필요 시점에 유연하게 대처할 수 있는 서비스 관점의 아키텍처 관리 개념 도입

IT 서비스 관리 환경 변화

비즈니스중심으로 TCO/ROI 전환

- IT 투자비용의 감소에 따른 기투자 자산의 재활용 및 투자 효율화
- 비용 효율성 확보 - 적은 비용으로 더 많은 서비스 제공
- 업무 확장성 및 개발 생산성 제고
- 지속적인 프로세스 모니터링/개선

IT기술 성숙과 시장 활성화 진입

- 분산 이기종 플랫폼간의 유연한 연계가 가능한 XML 표준 기반의 웹서비스 기술 발전
- 기술적 성숙도 및 기술표준 준수
- 표준기반의 관련된 개발 도구, 운영 플랫폼 지원 확장

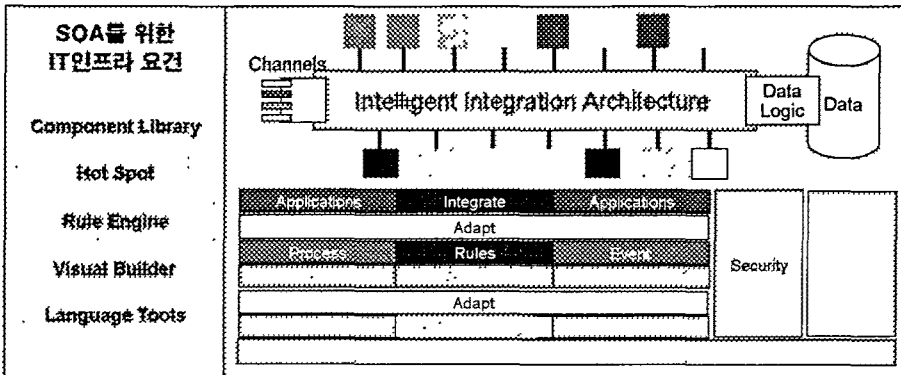
Max Soft

## SOA를 위한 IT인프라 요건 - Agility

변화하는 IT 환경과 고객의 요구를 빠르게 반영

- 요구 사항 : 신규 서비스의 창출, 운영 중인 서비스의 변경 및 수정, 솔루션 구조의 변경
- End-user가 현장에서 리얼타임으로 요구를 반영할 수 있도록 지원
- Platform 내에 S/W 생산 기술 및 저작 도구 지원

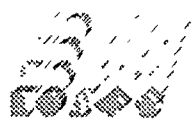
통합성, 재사용성, 확장성, 개발 편의성, 안정성, 관리 용이성 극대화



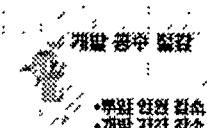
Max Soft

# SOA를 위한 IT인프라 요건 - Modularity & Productivity

- 기능 중심 Reuse
  - 소스 코드 대폭 축소
  - 모듈 재사용성 증대
  - 둘 시스템과의 통합 용이
  - 소스 가독성 및 유지 용이
- Rule engine 기반 개발



기능 모듈들의  
조립

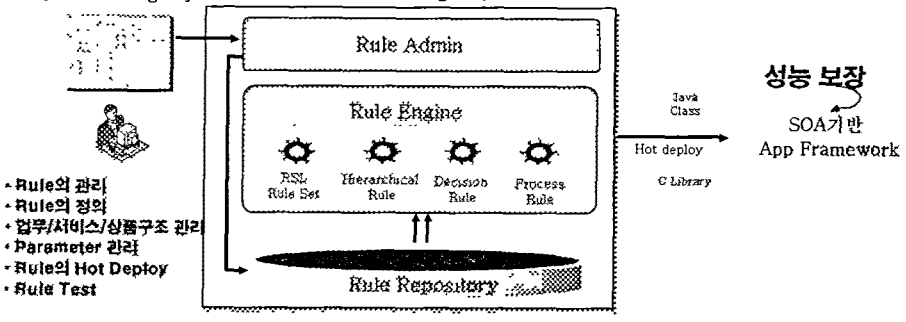


개발 생산성 향상

- 부원 안전 감소
- 개발 기간 감소

유지보수 용이  
[ Rule Manager ]

개발생산성 증가  
[ Rule Engine ]

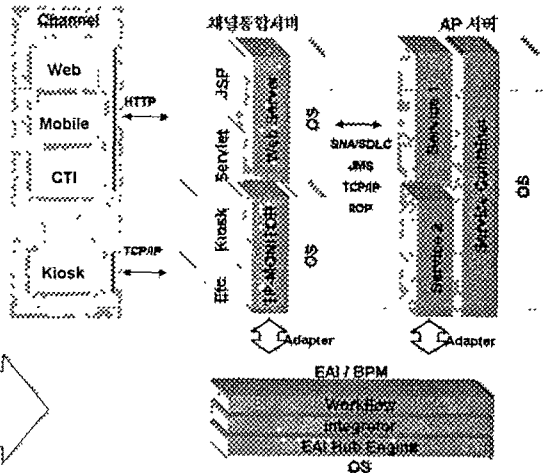


- Rule의 관리
- Rule의 정의
- 업무/서비스/상용구조 관리
- Parameter 관리
- Rule의 Hot Deploy
- Rule Test

# SOA를 위한 IT인프라 요건 - Integration & Automation

## Seamless Integration from Front-end to Back Office

- 채널 기능 강화
  - 클라이언트 request 처리
  - EAI와의 연동을 통한 타 시스템으로의 Data routing 및 업무 처리
  - 채널단에서의 고객접점 데이터 취합 및 서비스 강화
- Scalability, Maintainability
  - 새로운 채널 수용 및 사용자 증가에 따른 확장이 용이한 구조

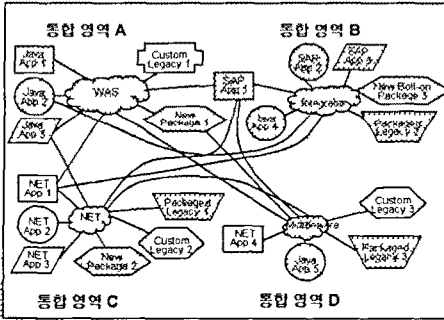


서비스에 직접 접근하고자 하는 고객들의 요구에 대응  
매체의 다양화에 따른 접근 채널의 증가 수용

# SOA를 위한 IT인프라 요건 - Real-time Architecture

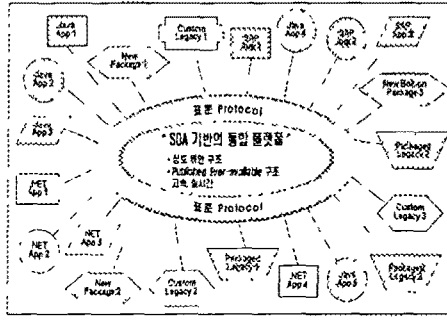
새로운 기술 구조에 대한 포용력이 있는 구조  
빠른 발전 속도에 발맞춰 갈 수 있는 구조

## 개별 아키텍처와 개별 통합



• Stove-pipe 구조

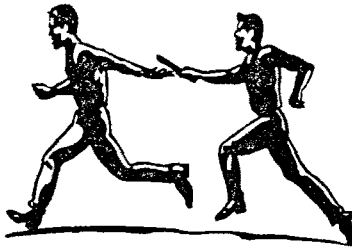
## 전사 통합 기반 시스템



- 상호 연인 구조
- Published, Ever-available 구조
- 고속, 실시간

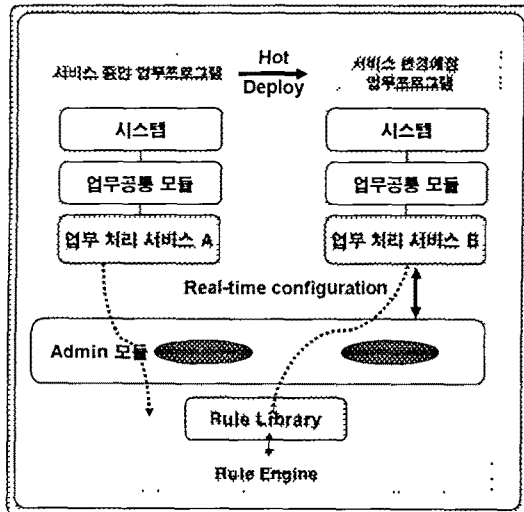
# SOA를 위한 IT인프라 요건 - Reflective Service Framework

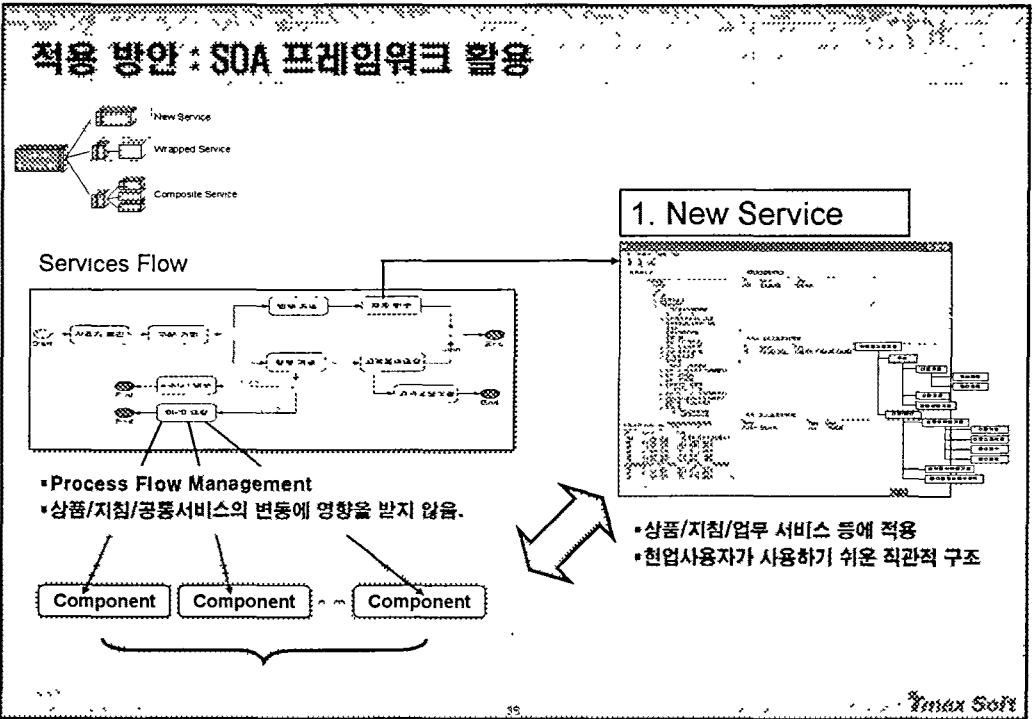
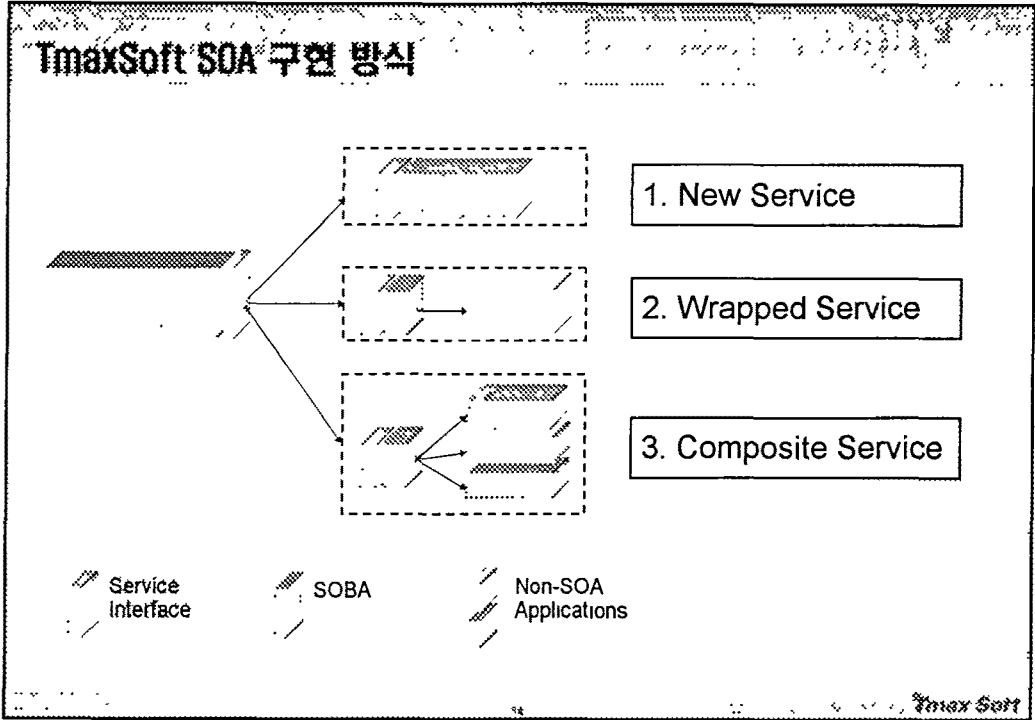
• Real-Time Enterprise 구현을 위한 Reflective 아키텍처로 구성된 Framework



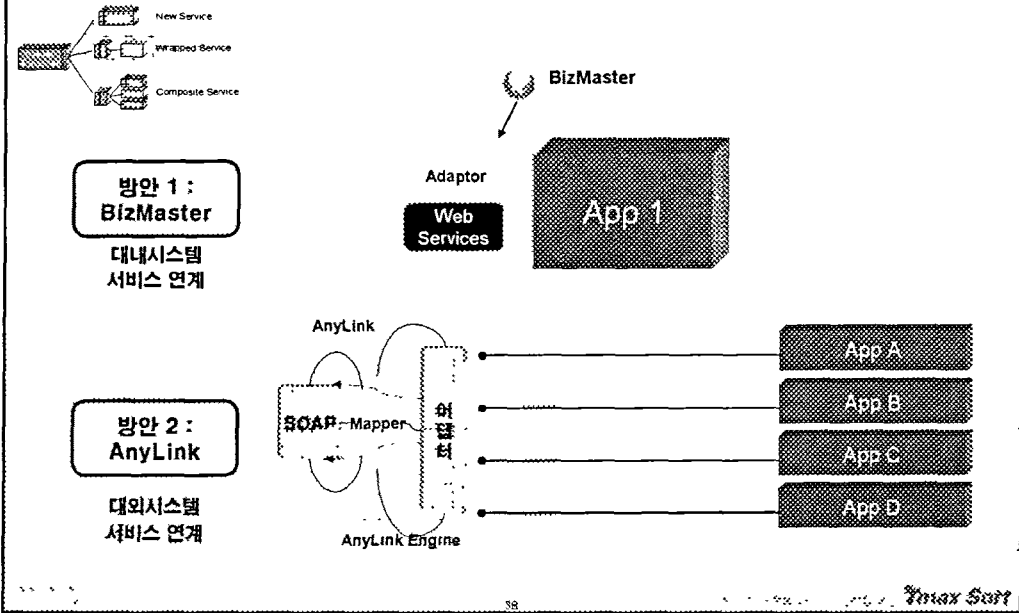
SOA기반의 RTE(실시간 기업)을 위한 Reflective Architecture

- Hot Deploy
- Real-time configuration 등

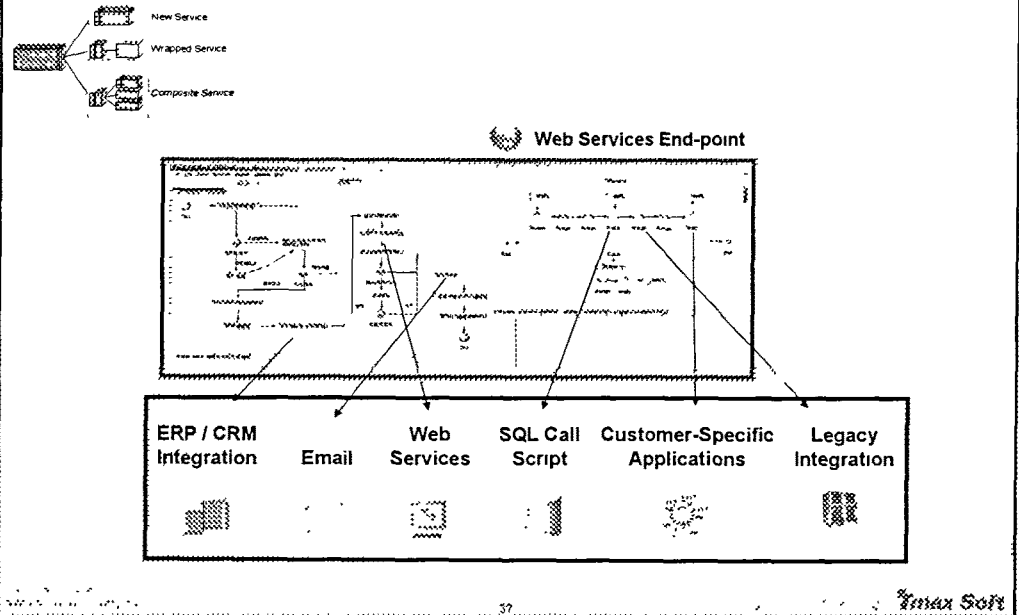




## 적용 방안 : Webservice기반의 EAI(BizMaster/AnyLink) 활용



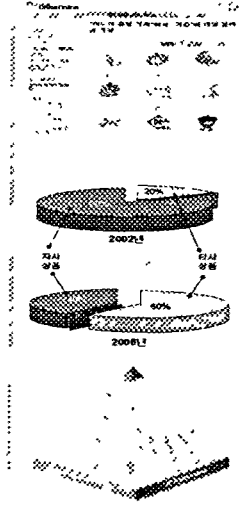
## 적용 방안 : BPM(BizMaster) 활용



# SOA와 프레임워크 개념 예시

MS-Office로 마케팅 보고서를 작성한다고 가정

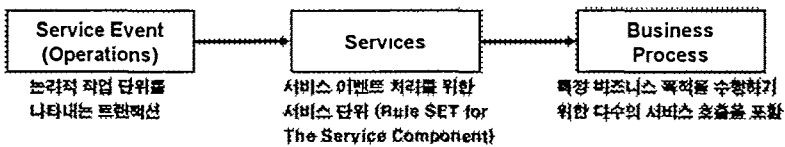
<b>Component</b>	<ul style="list-style-type: none"> <li>MS-Office에서 제공하는 클립아트 SET</li> </ul>
<b>시스템 컴포넌트 업무공통 컴포넌트</b>	<ul style="list-style-type: none"> <li>도형 서식 Set 및 마스터 (Story Board 포함)</li> <li>제목이나 Story Line 등의 Text 입력 필요</li> <li>실제 사용빈도 및 재사용 빈도 제일 높음.</li> </ul>
<b>Service Component (Rule Component)</b>	<ul style="list-style-type: none"> <li>자주 쓰이는 마케팅 주요 Template (시장점유율, 경쟁사 동향 등)</li> <li>마케팅 보고서에 핵심으로 사용, 재사용 빈도 높음.</li> </ul>
<b>프레임워크</b>	<ul style="list-style-type: none"> <li>여러 마케팅 보고서가 최적의 상태 (Best of Practice 형태)로 디렉토리화된 보고서 Set</li> </ul>



Tmax Soft

# TmaxSoft SOA 및 프레임워크 정의

## SOA Framework를 위한 Logical elements

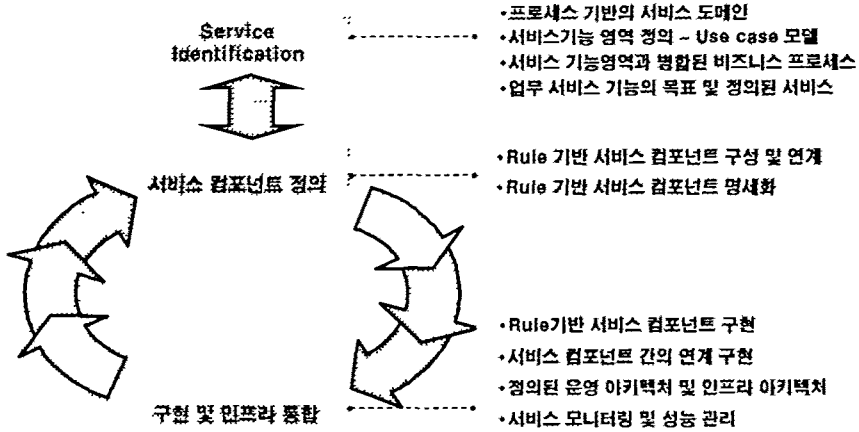


<b>TmaxSoft SOA 정의</b>	→ 표준 인프라 기반의 틀에서 서비스간 관계를 서비스/Rule 컴포넌트의 결합으로 구성하여, Process 중심의 서비스화를 지향하는 아키텍처
<b>SOA기반 Framework 정의</b>	→ Process 중심의 SOA기반 IT시스템 구축을 위한 분석, 설계, 구현, 테스트, 운영 단계의 Service Lifecycle을 Framework하에서 제공 → 서비스개발관점(Implementation View)과 운영관점(Operation View) 지원
<b>Framework하의 서비스 구현 단위</b>	→ 프로세스 중심 업무 서비스에서, 공동사용(재사용)이 가능한 서비스 컴포넌트 또는 업무로직 속성 Rule

Tmax Soft

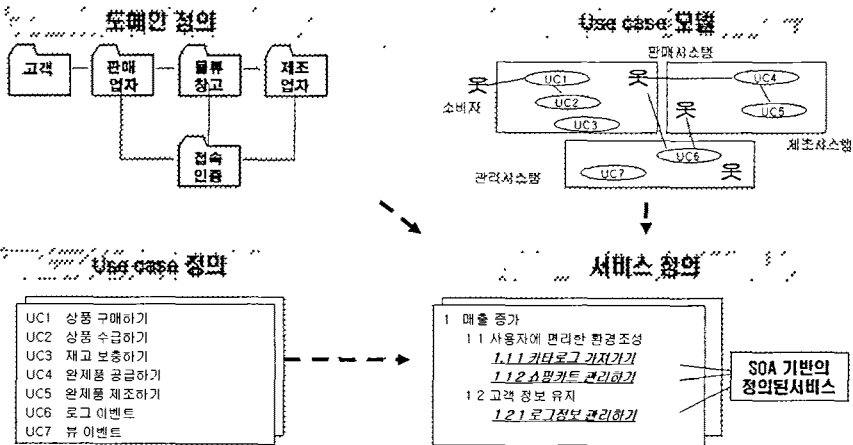
# SOA 기반의 개발 Approach

SOA의 개발 Framework은 비즈니스 관점에서 서비스를 정의 세분화하는 서비스 정의, 정의된 서비스를 어플리케이션화 하기 위한 컴포넌트 정의, 정의된 서비스와 컴포넌트를 기반기술과 연계하여 완성된 솔루션을 생성하는 구현 및 인프라 통합으로 구분할 수 있다.



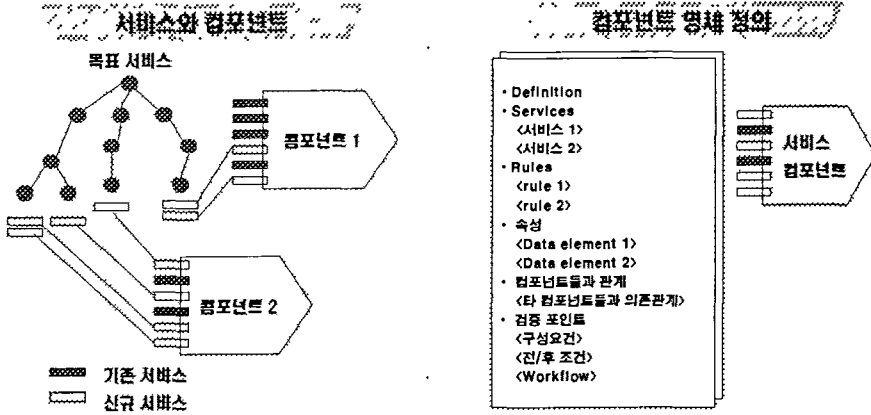
# 서비스 정의

서비스는 정의는 비즈니스를 효율적으로 지원할 수 있도록 비즈니스 관점의 도메인, Use case, 목표, 프로세스를 기반으로 작성되어 한다.



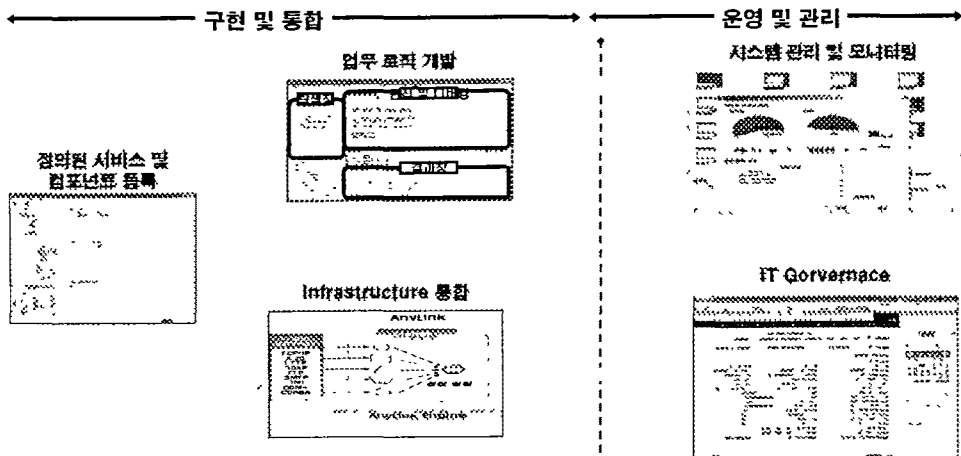
## 서비스 컴포넌트 정의

정의된 서비스는 기존의 서비스를 포함하는 컴포넌트를 구성하거나 필요시 신규 컴포넌트를 생성한다.  
컴포넌트 명세서에는 Rule, 서비스, 속성, 컴포넌트 의존관계, 검증 포인트 등이 포함 된다.



## 구현 및 인프라 통합

시스템화된 서비스 및 컴포넌트의 등록, 불필요한 코딩의 자동생성 지원, 통합된 인프라를 제공이 가능한 시스템을 통해 구축 서비스는 운영단계에서 지속적인 모니터링 및 관리를 통해 비즈니스에 미치는 악영향(Risk Mgmt.)을 최소화 하여야 한다.





# Benefits

## IT & Business Alignment

- 현업부서에 의한 비즈니스 설계, 모델링
- 프로세스 혁신과 모니터링
- 역할 및 책임 분담(Separation of Duty)
- 역할 세분화가 가능(Role Segmentation)
- 이상적인 전사 보안 정책 수립 및 운영 용이 (Security Enforcement)

## Ready for Next Generation

- Business logic & data can be anywhere - Location Independence
- On-demand / Adaptive Resource Management Support
- 프로그램/서비스 Deployment 실시간화
- RTE 지원의 현실화

## ROI 개선

- 기존 투자 보호
- 비용 효율성 확보
- Time-to-Market, 상품 경쟁력 확보
- 고객 차별화 서비스를 통한 고객만족 향상
- 프로세스상의 White space 제거
- 업무/개발 생산성 제고

## 기업문화 변화

- 직원 참여의 자발적 유도
- 성과속검/보상(CPM)의 객관화 및 합리화
- 직원 교육의 자동화 및 최소화
- 기업 혁신의 긴장도를 강화

Imax Soft

# ProFrame를 통한 SOA 구축 기대 효과 - 개발 패턴의 혁신

개발 비용 절감

### Framework

공통 모듈
Forms
Layout
Menu
Login
Calendar
ZipCode

### BL 개발자

개발 가이드

API 호출을 통한  
공통모듈 활용

개발 Template

테스트

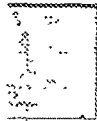
### 시사점

- 공통 모듈 재사용을 지원하는 틀이 없음
- 주로 UI 공통 모듈 위주
- SOA를 지원하기 위한 Service Grouping등의 기능 결여

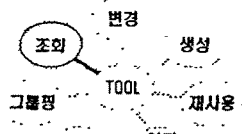
ProFrame

시스템 공통

업무 공통



유연한 공통모듈 생성



### Benefits

- 논리 아키텍처를 가시화하는 개발툴(ISE) 제공 (ProRule, ProBuilder)
- 업무 공통 모듈 재사용
- Macro Services 개발 지원
- SOA/EDA 완벽한 지원

Imax Soft

