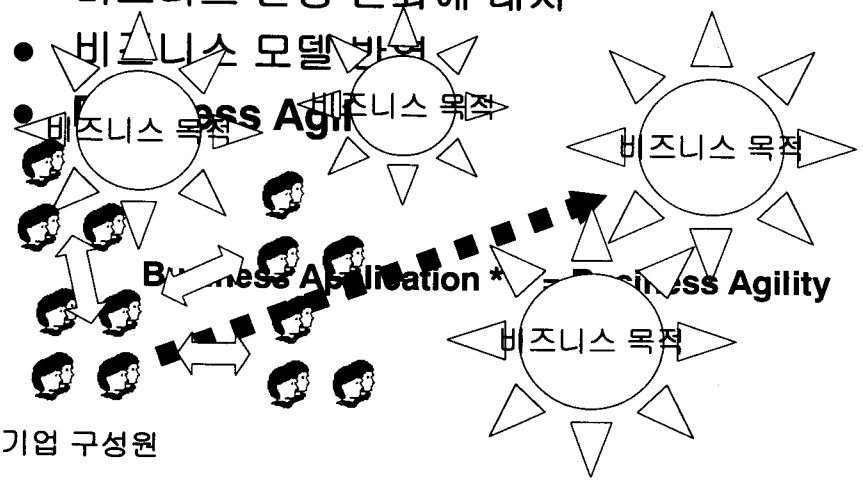
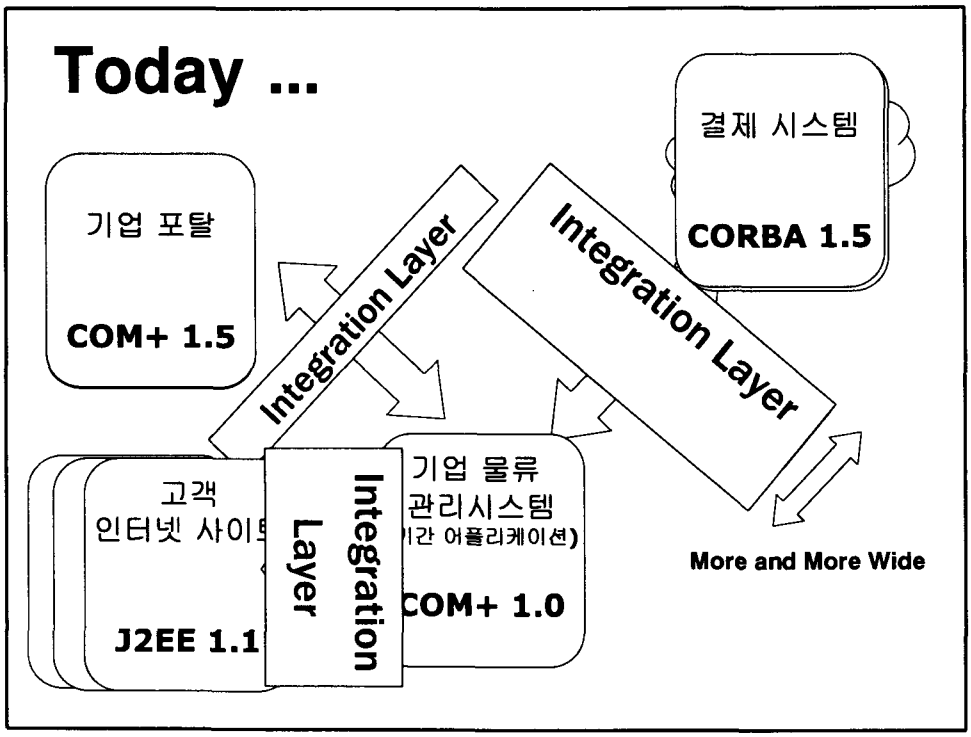
Service Oriented Architectures

이 철희 (chulh@microsoft.com)
Developer Tools Marketing Director
Microsoft

Enterprise Application

- 비즈니스 목적, 요구 사항 만족
- 비즈니스 환경 변화에 대처
- 비즈니스 모델 반영
- [illegible]ess Agility





Softeare Problems

- 통합은 기업 어플리케이션의 필수 요소
- 통합에 대한 예측이 힘듦
- 다양한 플랫폼으로 구성의 필연성
- 어플리케이션 통합의 복잡성

Software Solutions

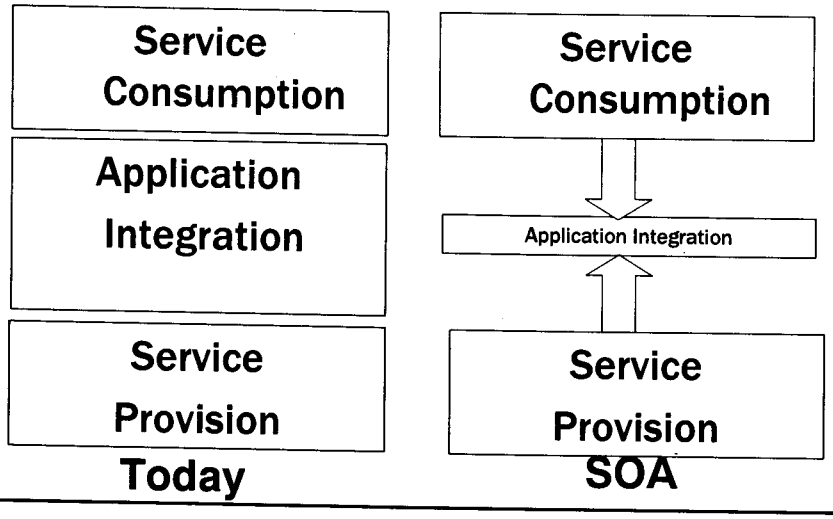
- 통합성, **Integration**
- 유연성, **Flexibility**

- 안정성, **Reliability**
- 보안성, **Security**
- ....

ON EVERY DEVICES

Bottom Line - SOA

- **Service Oriented Architecture**



분산 객체 기반 기술의 단점
Tightly Coupled Architecture

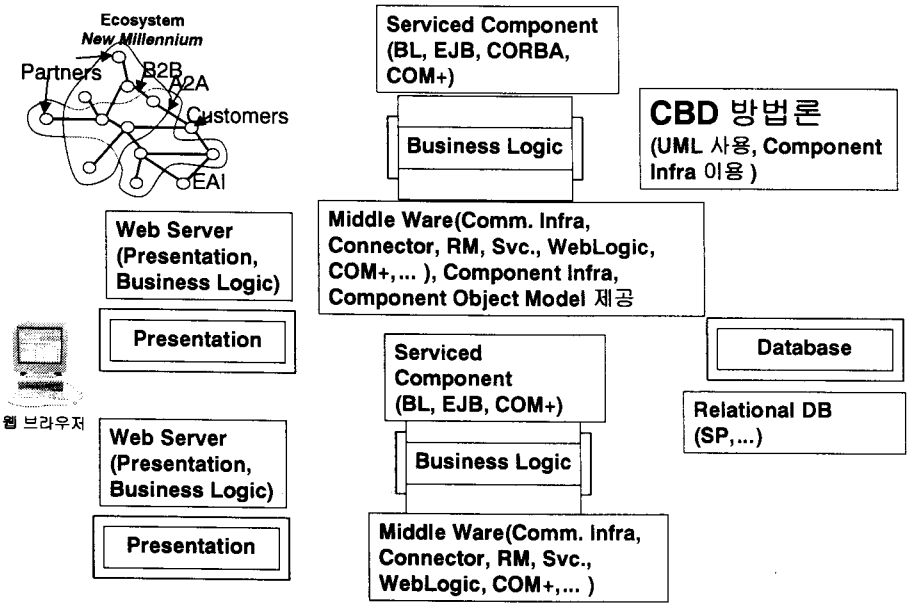
- 플랫폼, 벤더, 버전, 시간, 공간에 종속
- 변경 및 업데이트가 힘듦
- 확장 성이 떨어짐
- 인간 친화적이지 못하며 설계가 힘듦
- 유지 보수 비용이 증가
- 테스트가 힘들며 통합성이 떨어짐
- ...

문서 지향 기반 기술의 장점
Loosely Coupled Architecture

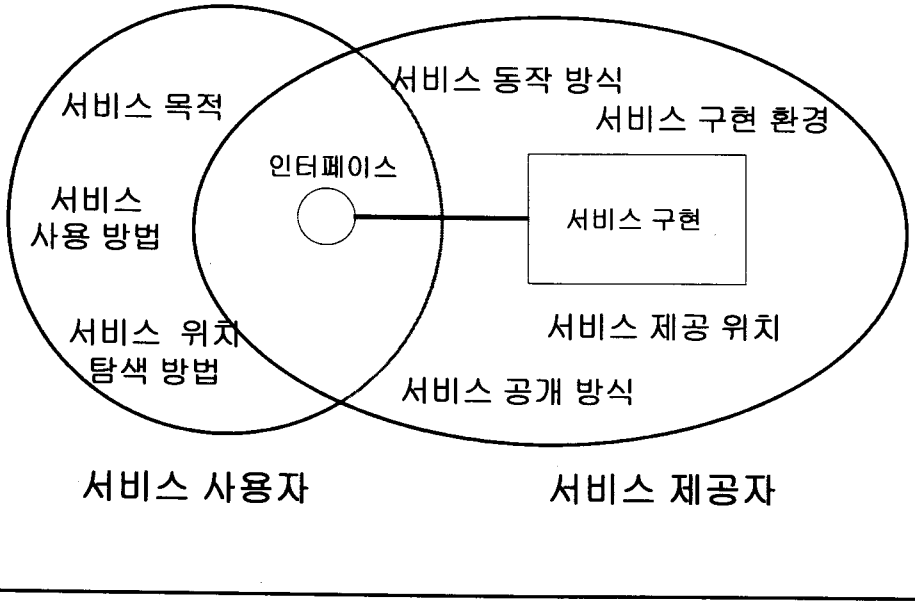
- 기술 독립적인 비즈니스 인터페이스
- 인터페이스 변경에 보다 유연하게 적응
- 적은 네트워크 라운드 트립 } 성능 향상 가능성
- 조정 가능한 캐싱, 머징 } 성능 향상 가능성
- 안정적, 스케일러블 서비스 가능
- 인간 친화적 어플리케이션 분석 설계
- 오피스 자원의 어플리케이션 수준 활용

기간 어플리케이션 재 사용 및 영속성 보장

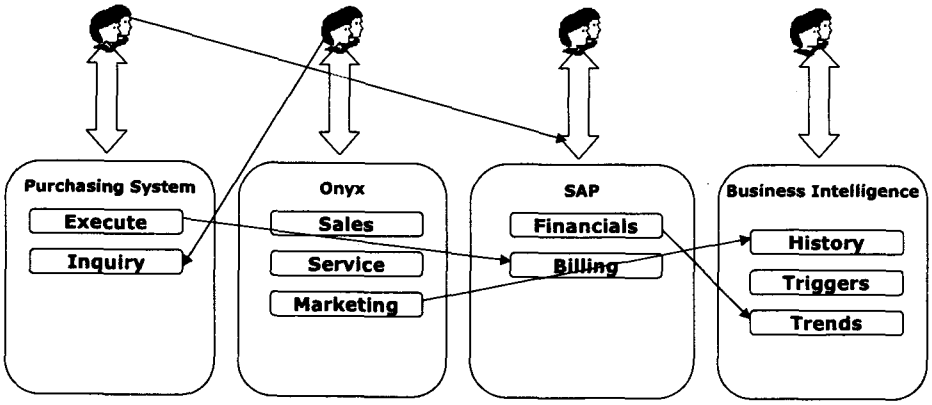
서비스 지향으로 진화



요구되어지는 아키텍처



Enterprise Integration will reach US$50 billion in 2003 - IDC -



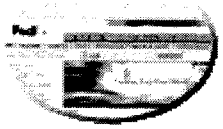
메시지 기반 기술 구현 예

?

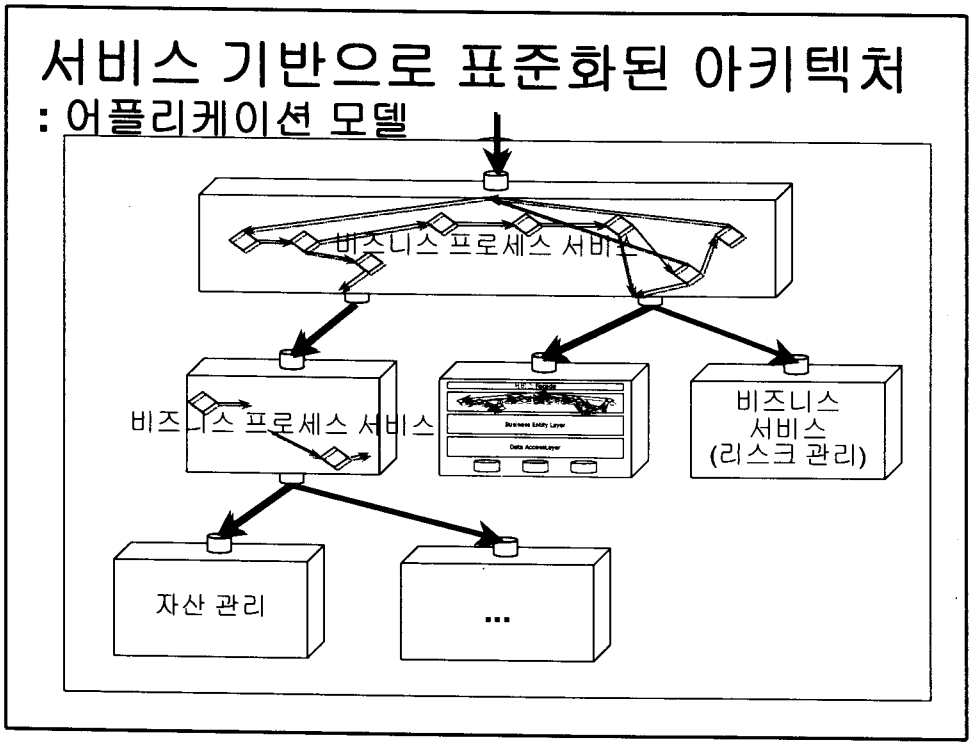
```
<?xml version='1.0'?>
  <Account Id="001" >
   <AccountDetail AccountId
   ="00A">
    <Customer Address="..."/>
  </AccountDetail>
  </Account>
```

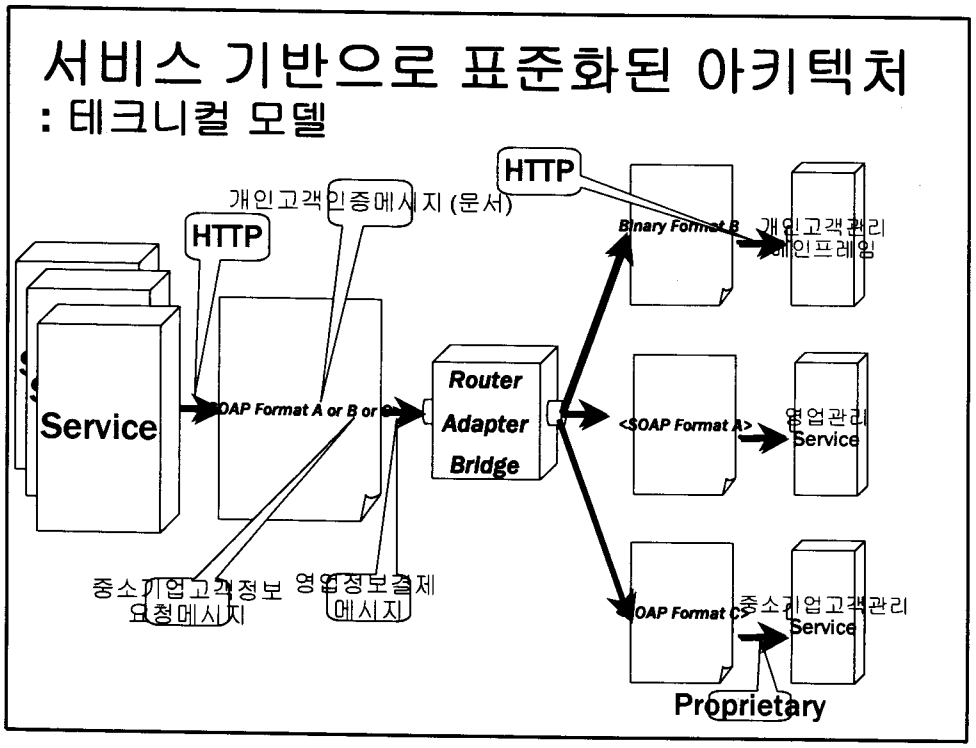
```
<?xml version='1.0'?>
  <ProcessResult
  Id="001">
   <Tracking
  Num="00AF"/>
  </ ProcessResult >
```

- 플랫폼 독립적
- 인터페이스 변경 적응력
- 서로 다른 표준 지원
- 비즈니스 인터페이스



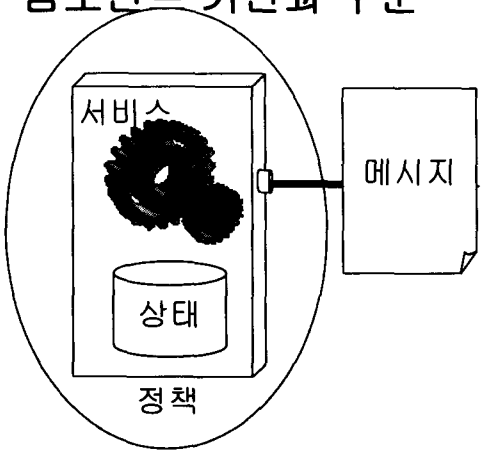
여신 관리 시스템


서비스 기반으로 표준화된 아키텍처
: 어플리케이션 모델
비즈니스 프로세스 서비스
비즈니스 프로세스 서비스
비즈니스
서비스
(리스크 관리)
자산 관리
...


서비스 기반으로 표준화된 아키텍처
: 테크니컬 모델
HTTP
개인고객인증메시지 (문서)
HTTP
Binary Format B
개인고객관리
레인프레임
Service
SOAP Format A or B or C
Router
Adapter
Bridge
<SOAP Format A>
영업관리
Service
중소기업고객정보
요청메시지
영업정보결제
메시지
SOAP Format C
중소기업고객관리
Service
Proprietary

메시지 기반 어플리케이션

- 메시지 기반 기술을 이용하는 서비스 기반 개발
- 컴포넌트 기반과 구분



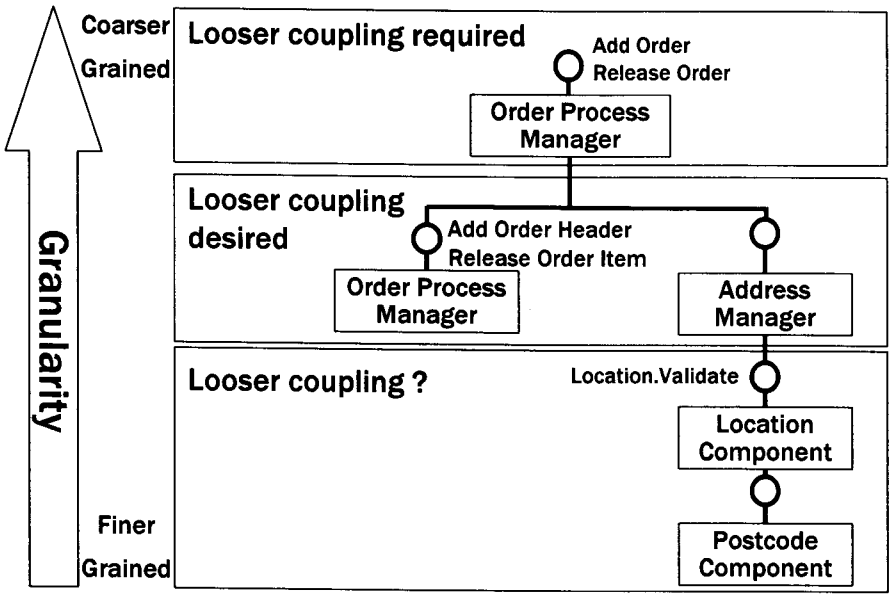
서비스 vs. 컴포넌트/객체

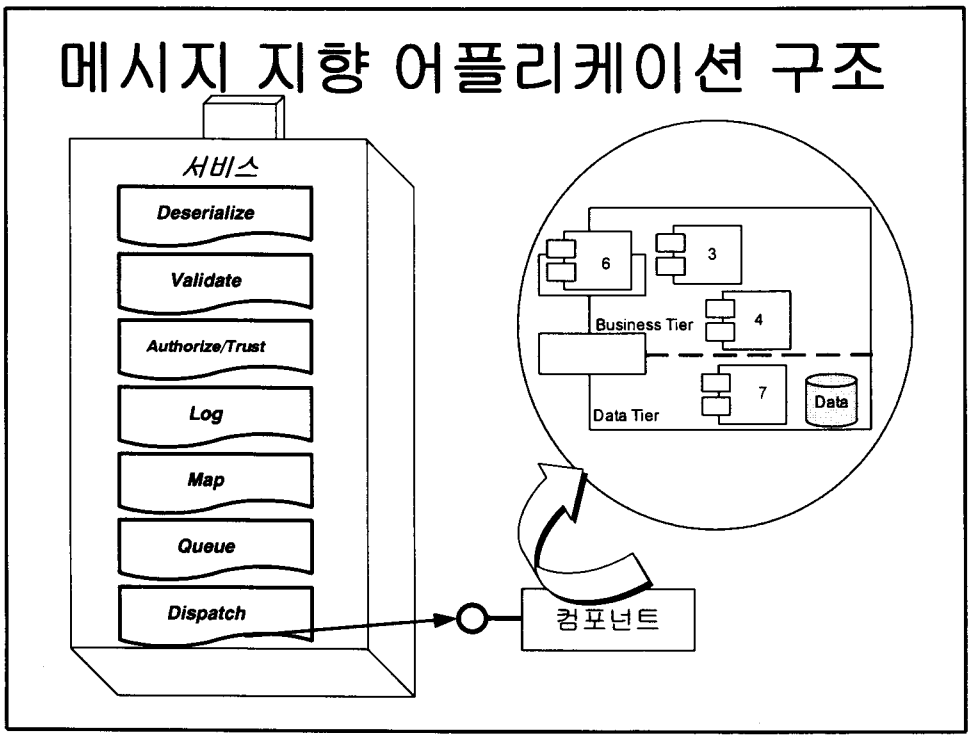
- 컴포넌트
  - 생성 모델 **(instancing model)** 이 있음
  - 컴포넌트/객체의 인터페이스는 메쏘드 시그너처를 정의 **(e.g. IDL)**
  - 컴포넌트내의 객체 생성 후, 인터페이스에서 정의한 메쏘드 호출
- 서비스
  - 생성 모델이 없음
  - 주어진 주소로 메시지를 전달
  - 서비스 인터페이스는 메시지의 형태를 정의 **(e.g. WSDL)**

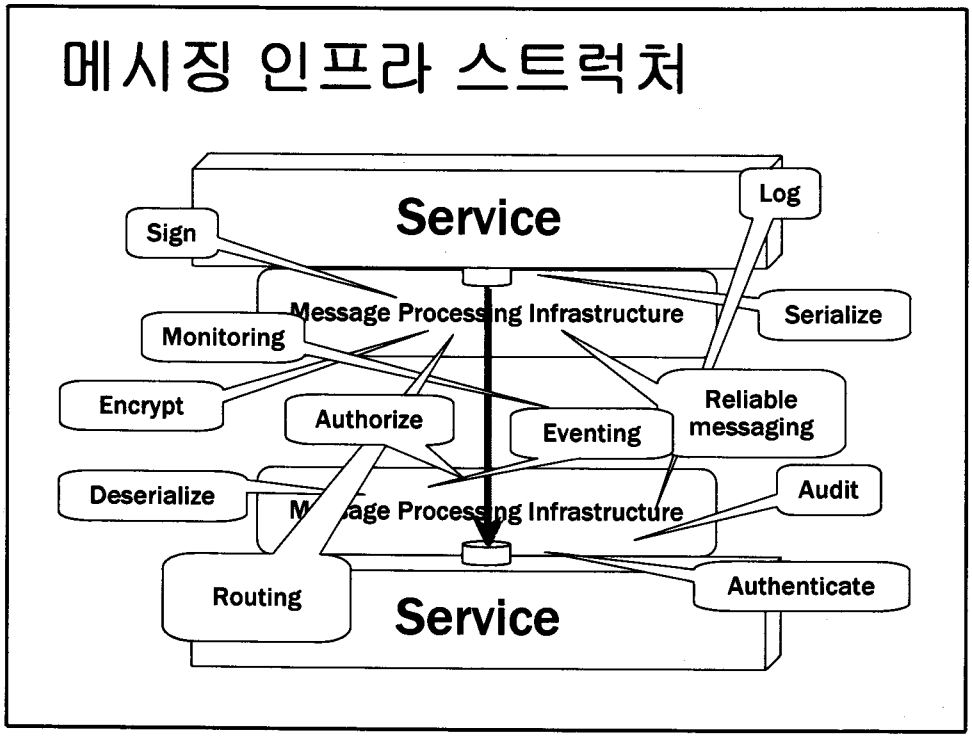
메시지

- 서비스는 메시지를 이용해 상호작용 함
- 메시지 형태만 규약
  - 다른 플랫폼 간 통신이 가능
  - 플랫폼 교체가 가능
  - 새로운 서비스와 연동이 쉬움
  - **Loosely Coupled Architecture**

컴포넌트 or 서비스

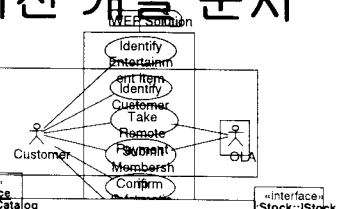



메시지 지향 어플리케이션 구조
서비스
Deserialize
Validate
Authorize/Trust
Log
Map
Queue
Dispatch
6
3
Business Tier
4
7
Data Tier
Data
컴포넌트

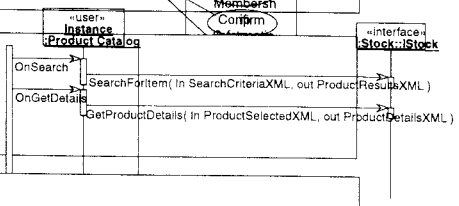

메시징 인프라 스트럭처
Service
Log
Sign
Message Processing Infrastructure
Serialize
Monitoring
Encrypt
Authorize
Eventing
Reliable messaging
Deserialize
Audit
Routing
Authenticate
Service

서비스 지향 어플리케이션 개발 순서

비즈니스 도메인 분석



비즈니스 서비스 버스 설계



메시지 설계 : ***XML, XSD, SOAP, WSDL,*** *기타 확장 표준*

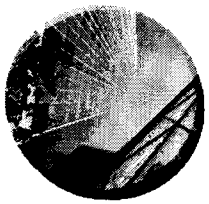
구현 : **.NET, WSDK, SOAP ToolKit**

퍼블리싱 : ***UDDI***

XML Web Service 의 장점

User: Beyond browsing

Business: Integration by design



Web services movement will be a turning point ...

- Tim Berners-Lee -

XML Web Services

Industry Standards for Interoperability

- **Enable disparate systems to work together**
  - **Across languages, platforms, applications**
  - **Computer to computer**
  - **Inside/outside the firewall**
- **Based on public, internet standards**
  - **XML, SOAP, WSDL, UDDI**
- **Broad industry support**
  - **Key area of vendor alignment**

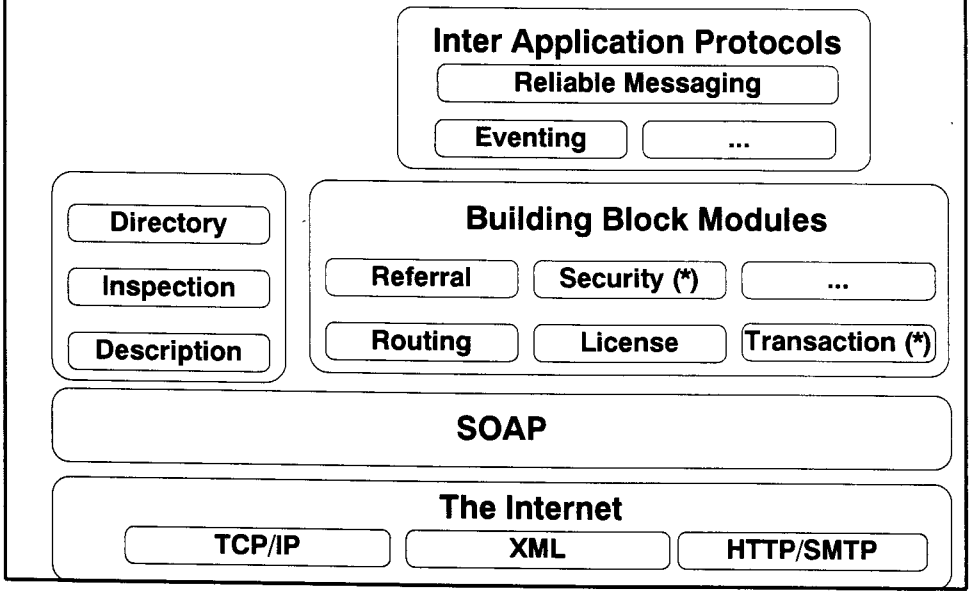
웹 서비스 표준 스택

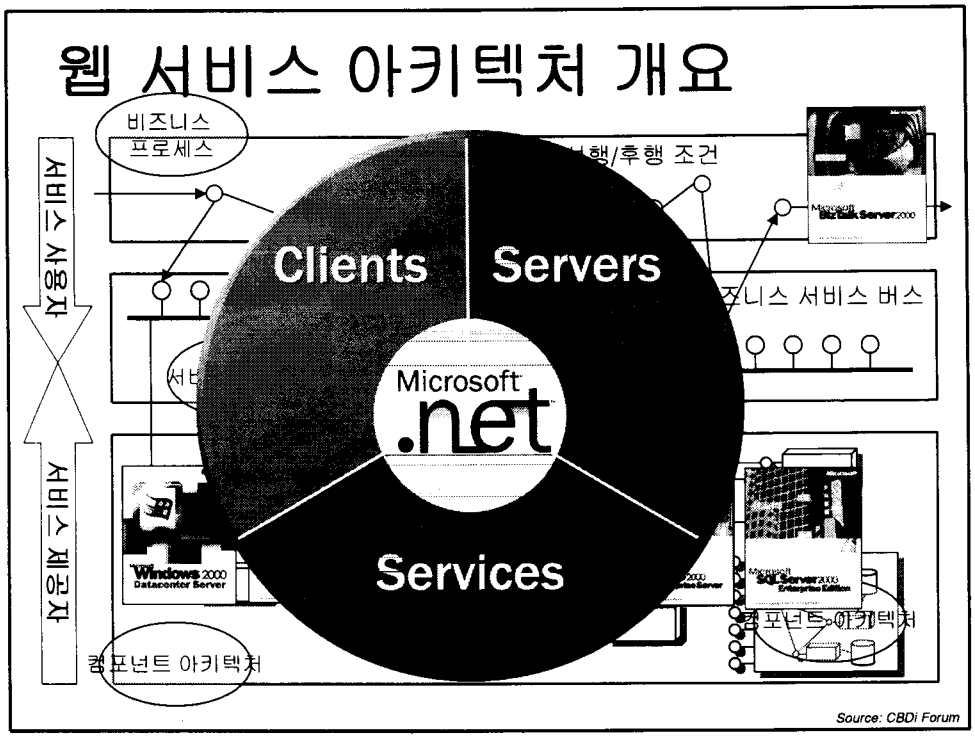
Wire Stack	Description Stack	Discovery Stack
Other	Business Process Orchestration	
Attachments / Routing	Message Sequencing	
Security / Reliability	Service Configuration Capabilities	
XML Protocol: Envelope and Extensibility (SOAP)	Service Description (WSDL)	Directory (UDDI)
Syntax (XML)	XML Schema	Inspection

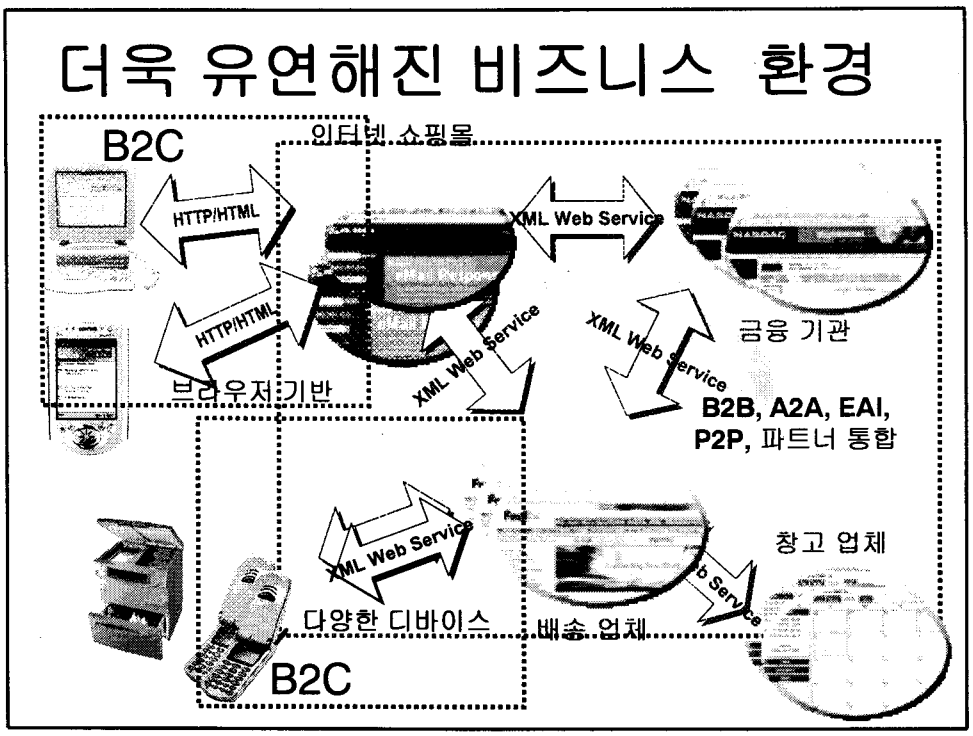
확장된 표준	업계 표준	확장 표준

Source: W3C Web Services Workshop

GXA, 웹 서비스 확장 프로토콜




웹 서비스 아키텍처 개요
비즈니스 프로세스
서비스 사용자
서비스 제공자
Clients
Servers
Services
Microsoft .net
행/후행 조건
조니스 서비스 버스
컴포넌트 아키텍처
컴포넌트 아키텍처
Source: CBDi Forum


더욱 유연해진 비즈니스 환경
B2C
인터넷 쇼핑몰
HTTP/HTML
HTTP/HTML
XML Web Service
XML Web Service
XML Web Service
브라우저 기반
금융 기관
B2B, A2A, EAI,
P2P, 파트너 통합
XML Web Service
다양한 디바이스
B2C
배송 업체
창고 업체

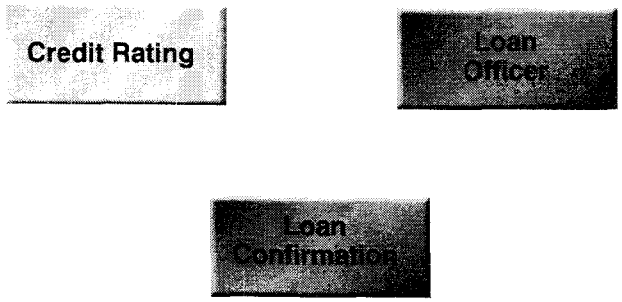
Summary

SOA로의 이동(Shift)

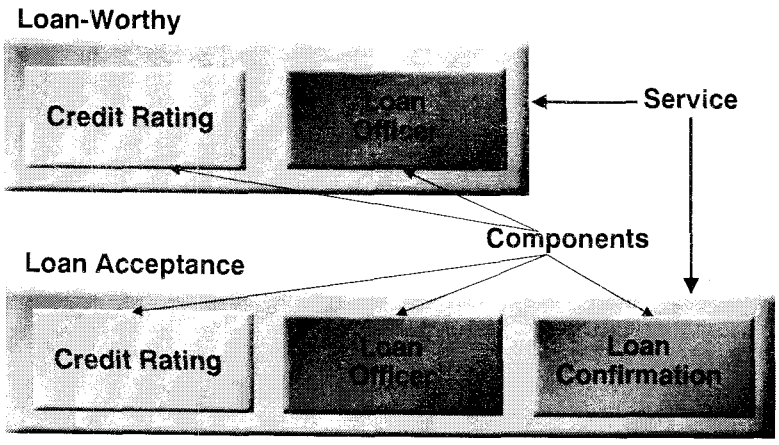
From	To
• Function oriented • Build to last • Prolonged development cycles	• Process oriented • Build to change • Incrementally built and deployed
• Application silos • Tightly coupled • Object oriented • Known implementation	• Orchestrated solutions • Loosely coupled • Message oriented • Abstraction

왜 SOA인가?

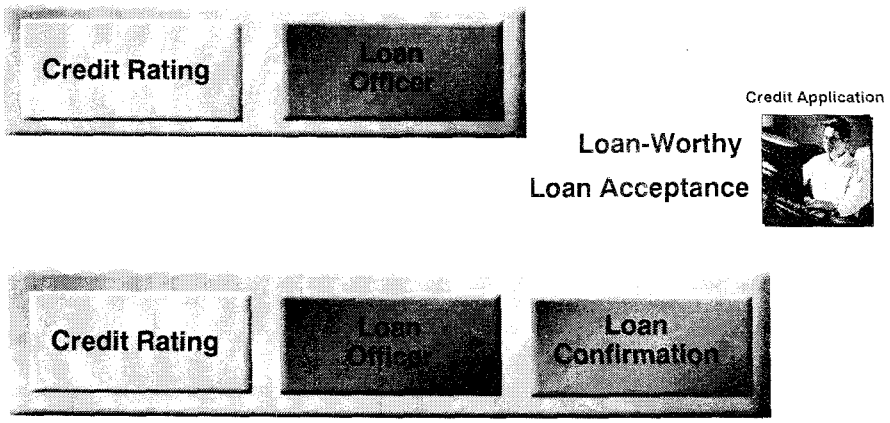
Current Component Library (3 Binaries)



Service (Composite Services)



Client Application requests two services



Client Application requests two services