

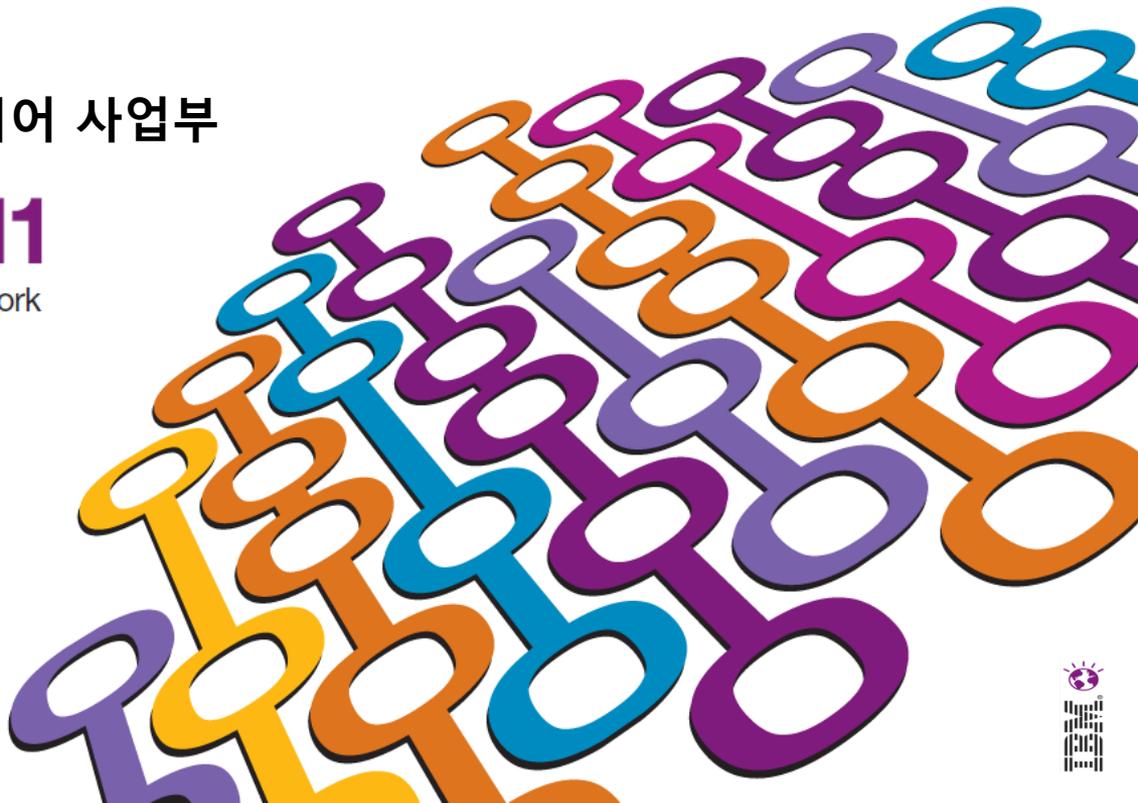
차별화된 EAI Solution IBM WebSphere Message Broker

조성현 차장

한국IBM 소프트웨어 그룹 웹스피어 사업부

Impact Korea 2011

Changing the Way Business and IT Leaders Work



오늘 논의할 내용은

1

EAI

2

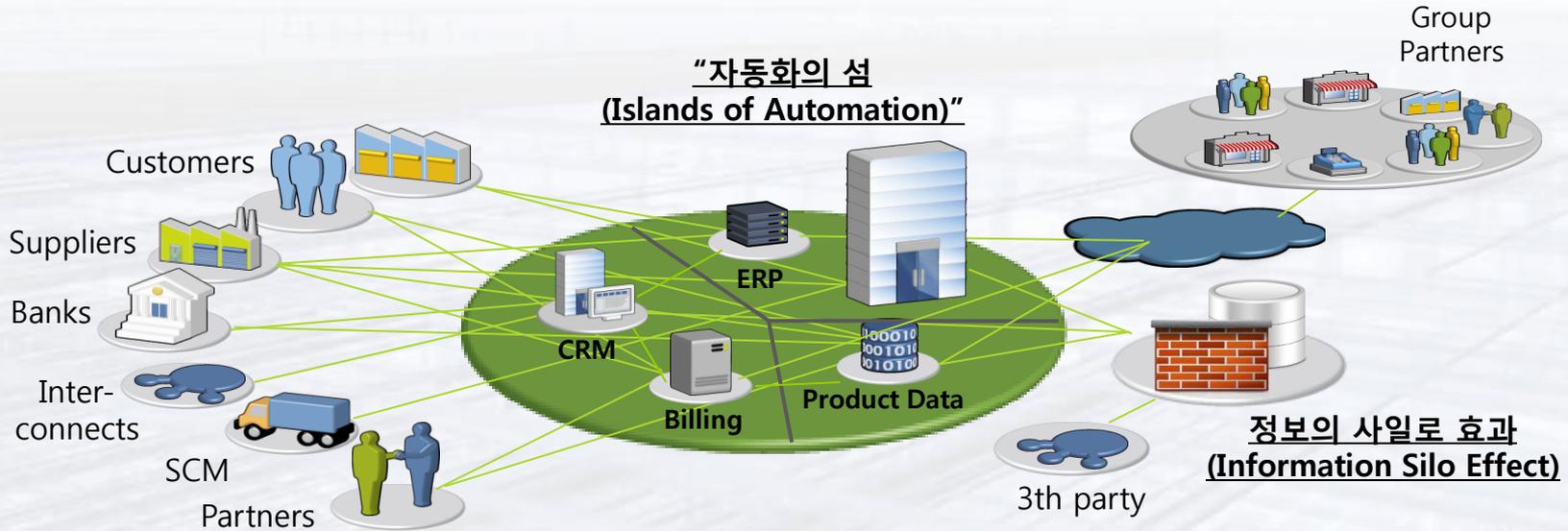
IBM WebSphere Message Broker

3

J2EE 기반의 EAI 솔루션과의 차별성



EAI (Enterprise Application Integration)





EAI (Enterprise Application Integration)



• EAI의 목적

1. Data & Process Integration (Enterprise Information Integration: EII)
2. Vendor Independence
3. Common Façade: common I/F





EAI 솔루션 정의 - By Gartner

2. Middleware Platform

유연한 확장성, 장애에 대한 대응력 및 신뢰성 있는 메시지 전달을 책임지는 인프라 제공 능력



5. Business Workflow

비즈니스 프로세스 또는 메시지 플로우를 설계, 운용 및 모니터링 하는 능력

1. EAI Solution provider or Vendor

EAI 솔루션 판매와 프로젝트 수행 및 지원을 일관되게 수행하는 기업 능력

3. Data Broker

애플리케이션 상호 간 데이터의 포맷 등의 차이에 대한 문제 해결 능력



4. Adapter

다양한 패키지 애플리케이션 및 레거시 시스템과 신속한 연결이 가능해야 하며, 재사용이 높은 인터페이스 지원 능력





WebSphere Message Broker



Universal Connectivity FROM anywhere, TO anywhere

- 프로토콜, 메시지 포맷 및 Mediation 패턴을 제공함으로써, 애플리케이션 간의 연결 단순화 제공
- 다양한 기술과 시스템에 간편한 연결성 제공



Protocols, Transformation, Data Format

- 광범위한 프로토콜 지원
예) MQ, TCP/IP, FTP, HTTP(S), 웹서비스 등.
- 다양한 형식의 데이터 포맷 지원
예) Binary(C/CORBOL), XML, Industry (SWIFT, EDI, HL7), 사용자 정의 포맷 등.



High Performing, Scalable & Managed

- 광범위한 OS 및 하드웨어 플랫폼 지원
- 전통적으로 대용량 트랜잭션 환경 지원
- 실시간 전개/변경, 메시지 플로우 테스트, 원격 제어 등 다양한 관리 옵션 제공
- 수평 & 수직 확장성(Scalability) 제공



Simple Programming & Productivity

- 쉽게 개발하고 쉽게 관리하는 편의성 제공
: 쉽고 빠르게 배우고, 개발하고, 테스트 가능
- Top-Down 방식의 개발 템플릿(패턴) 제공
- 다양한 개발 언어 지원
예) Graphic Mapping, Java, ESQL, PHP, XSLT 등

IBM WebSphere Message Broker는 1998년부터 국내외의 다양한 애플리케이션 통합 영역에 도입, 적용되어 온 검증된 EAI 솔루션입니다.



Extensive transformations of data formats



WMB: 솔루션 컴포넌트 아키텍처

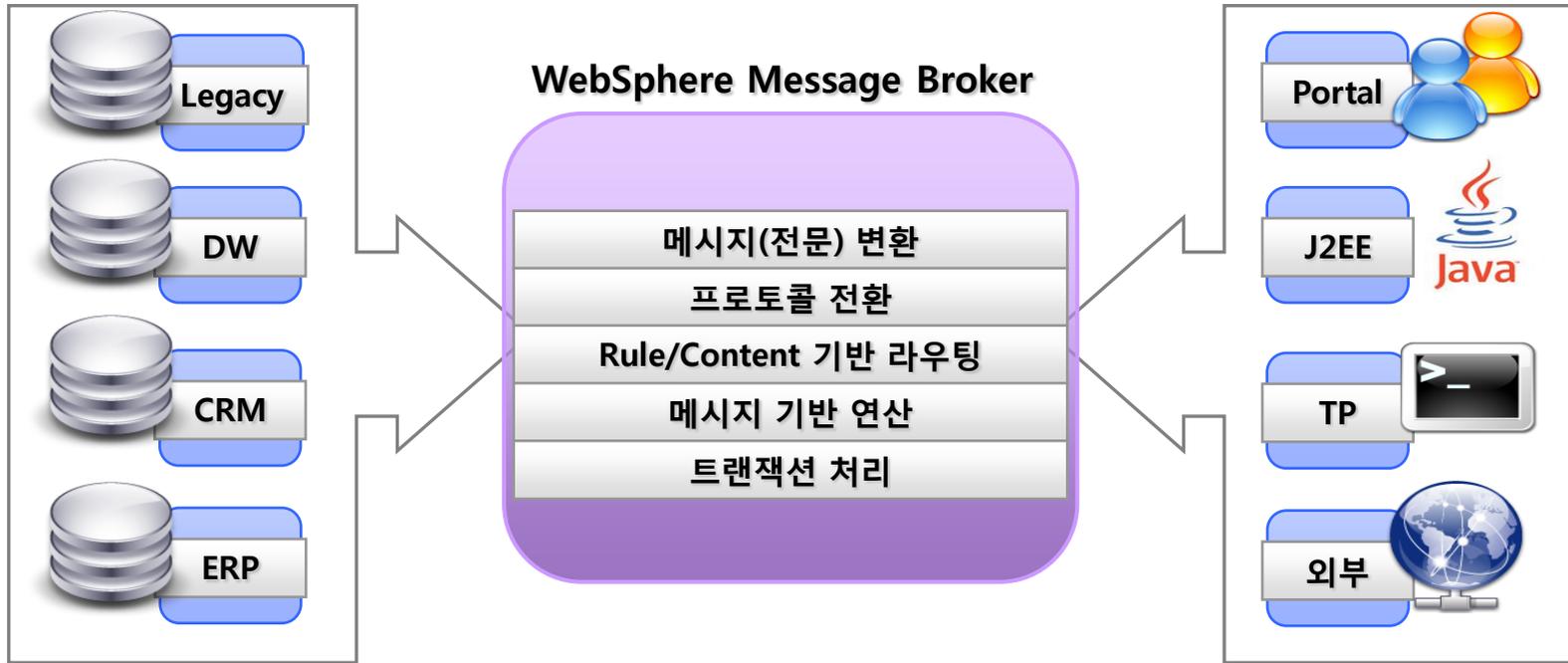
솔루션을 최대한 간결하게 유지하기 위해 반드시 필요한 Pre-requisite 요소 최소화





IBM EAI 솔루션: WebSphere Message Broker

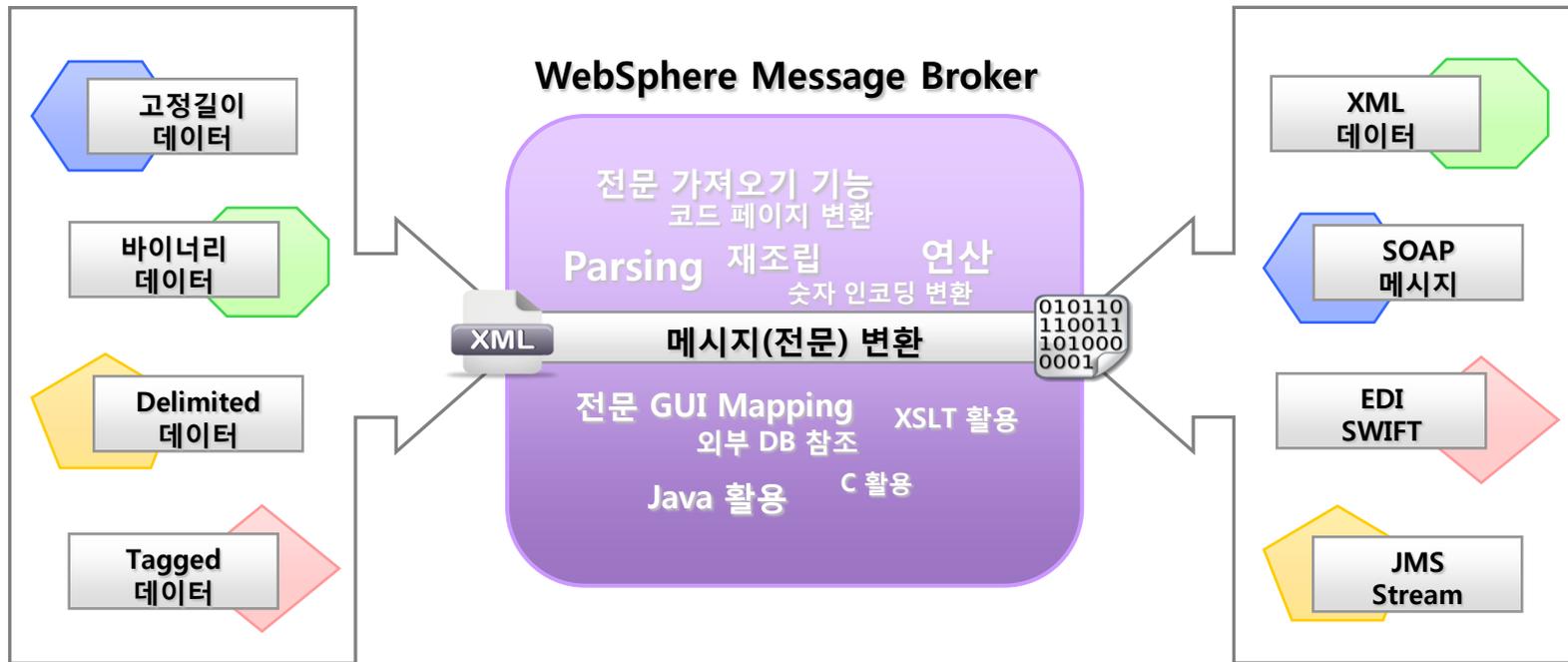
MQ 기반의 Native EAI 솔루션



WebSphere Message Broker: Any to Any 전문 변환



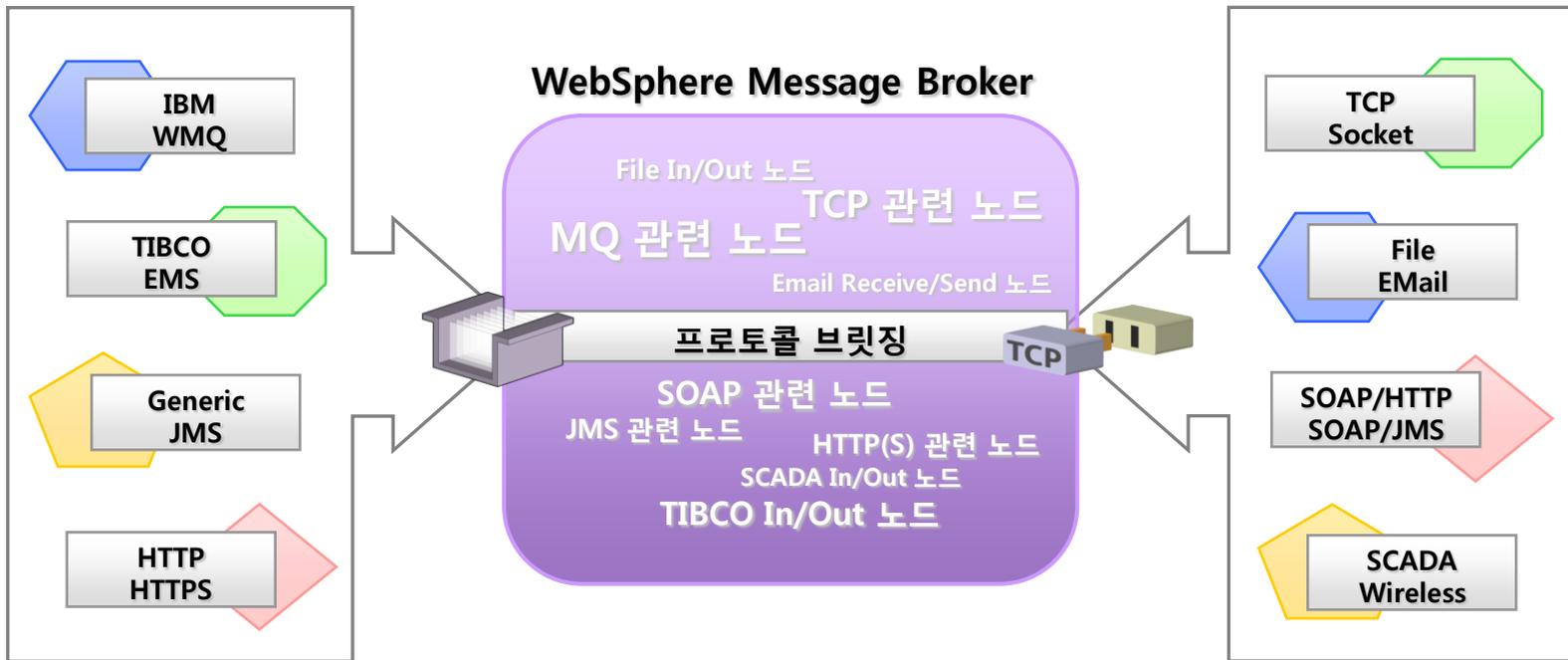
기존 애플리케이션의 변경을 최소화하면서 기업 내에 존재되어 있는 전문을 일관된 방법으로 변환



WebSphere Message Broker: 프로토콜 변환

다양한 데이터 전송 프로토콜 지원 및 프로토콜간의 변환 제공

데이터 송수신 방식을 사용하는 기존 애플리케이션 변경을 최소화하는 동시에 통합을 수행하는 기반 제공



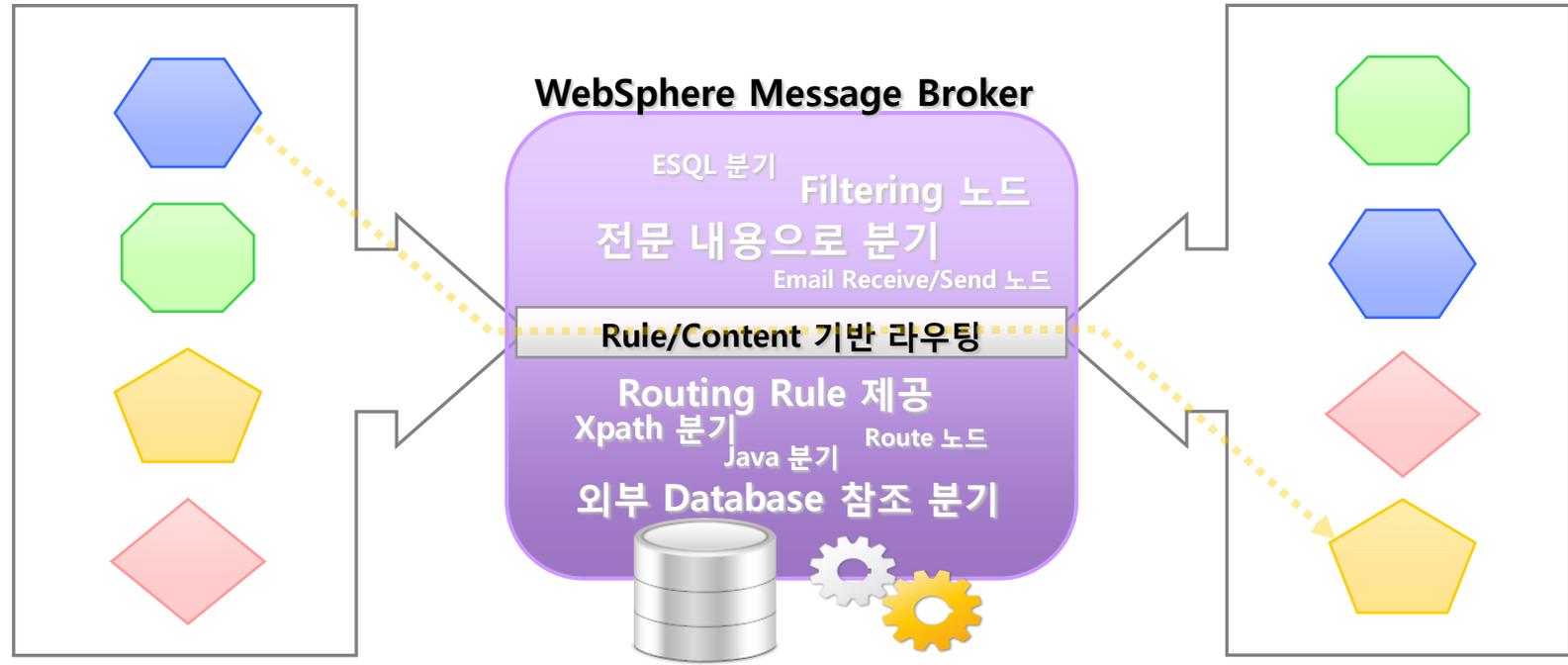
*노드: WebSphere Message Broker 내의 로직을 구현할 때 제공되는 기능 단위





WebSphere Message Broker: 룰/콘텐츠 기반 라우팅

전문 메시지를 동적/정적으로 하나 또는 다수의 목적지로 분기하는 룰/콘텐츠 기반의 라우팅 기능 제공
기업 내 애플리케이션 간 통합의 경로나 로직의 변화에 적절하게 대응 및 기존 애플리케이션의 변경 최소화



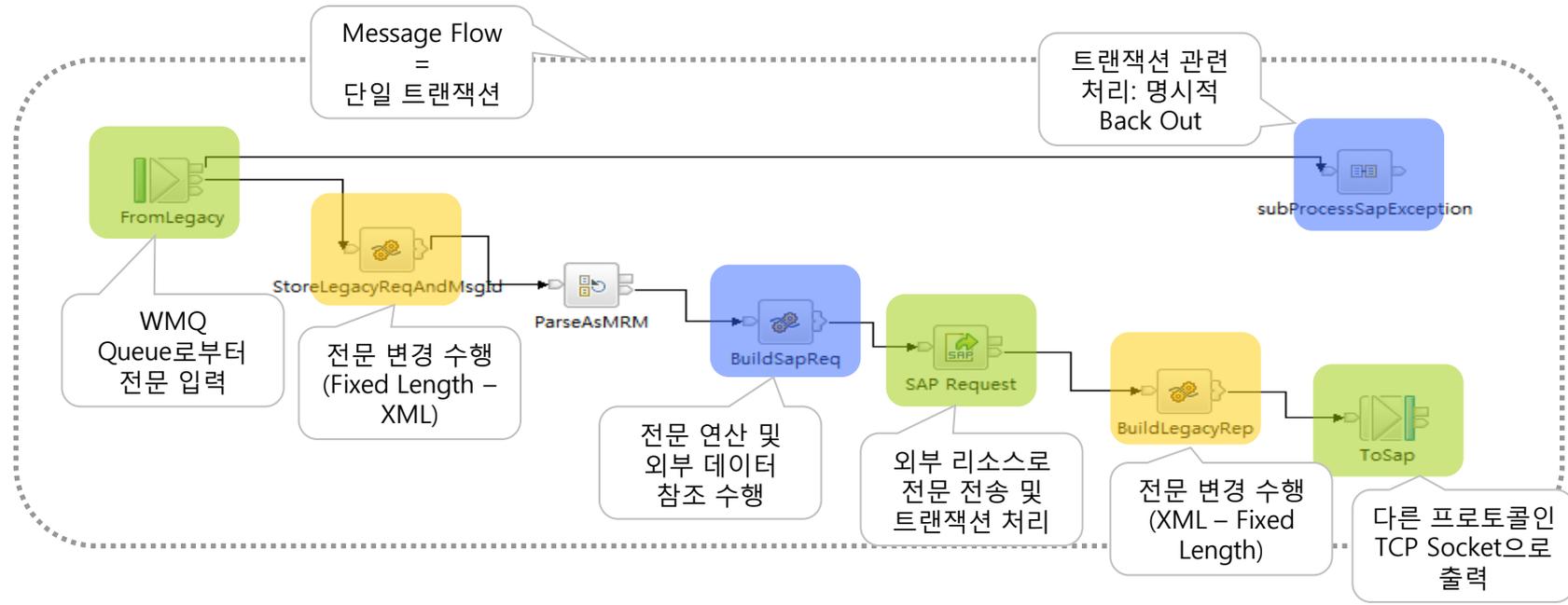
*노드: WebSphere Message Broker 내의 로직을 구현할 때 제공되는 기능 단위





WebSphere Message Broker: 중개 로직, 메시지 플로우

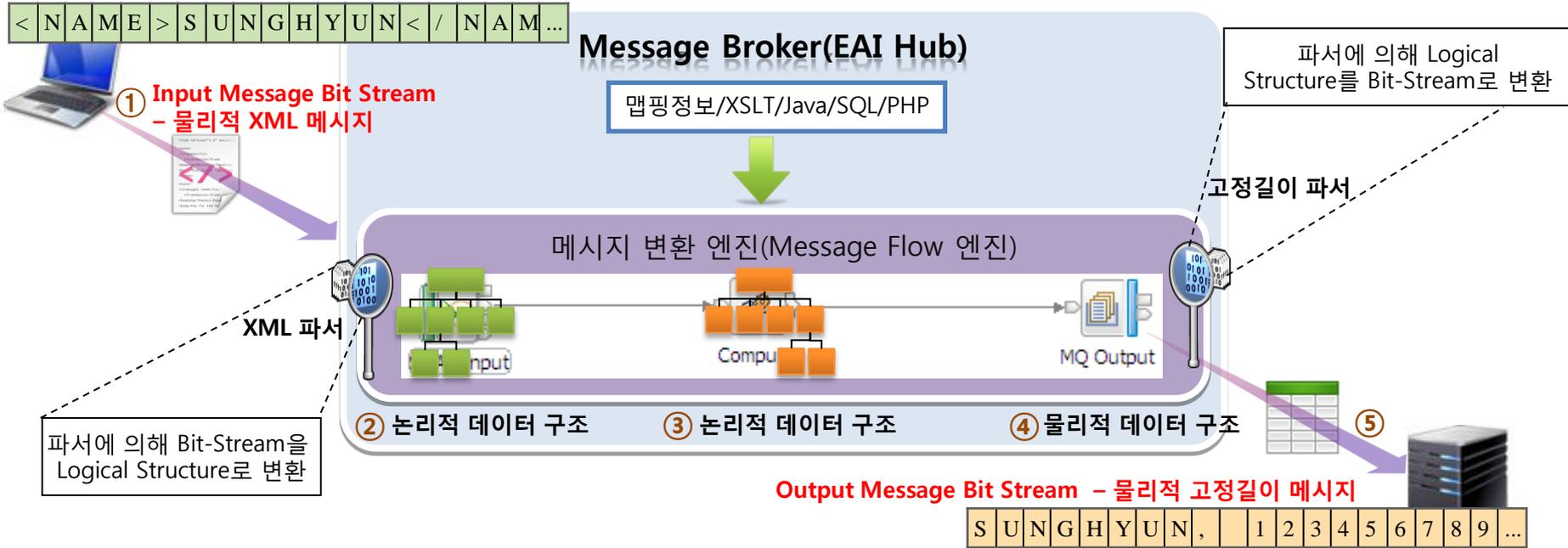
애플리케이션 간의 전문 메시지를 중간에서 받아 파싱, 변환, 프로토콜 전환, 연산, 트랜잭션 처리 등 수행





WebSphere Message Broker: 파싱 및 메시지 변환

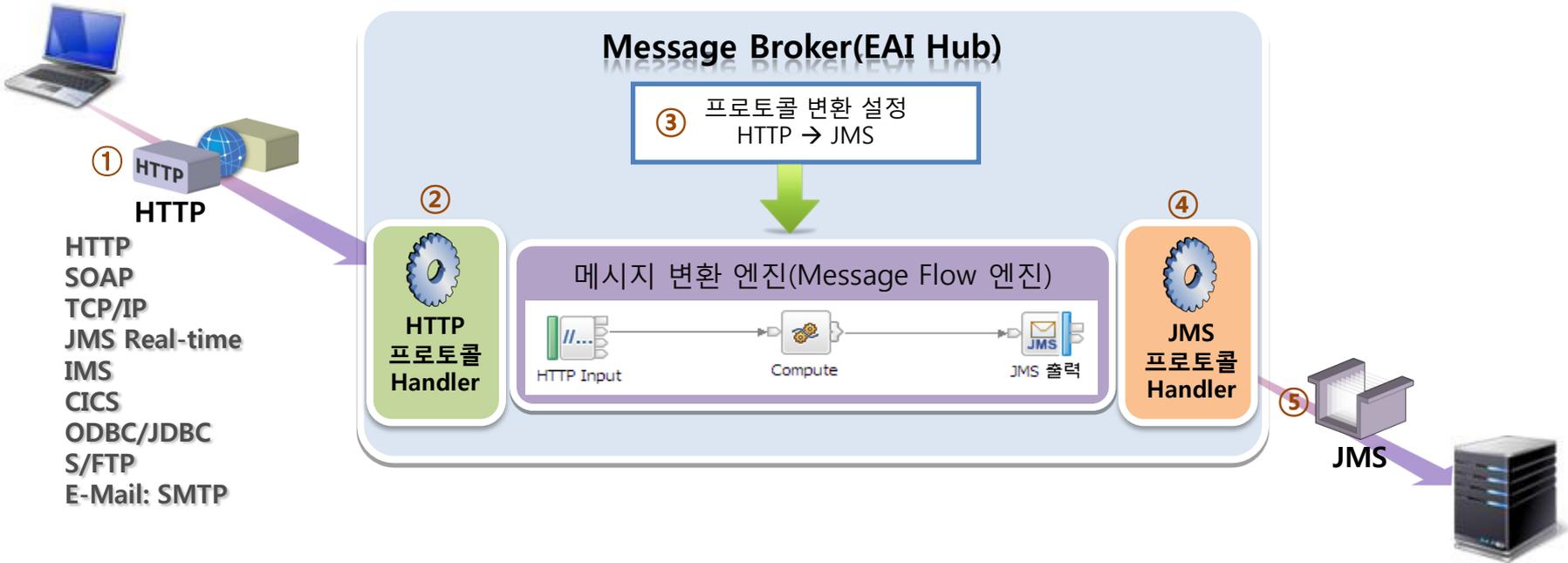
메시지 플로우 내부에서 다양한 형식의 메시지 데이터를 인식 및 가공하는 방법 제공





WebSphere Message Broker: 프로토콜 변환

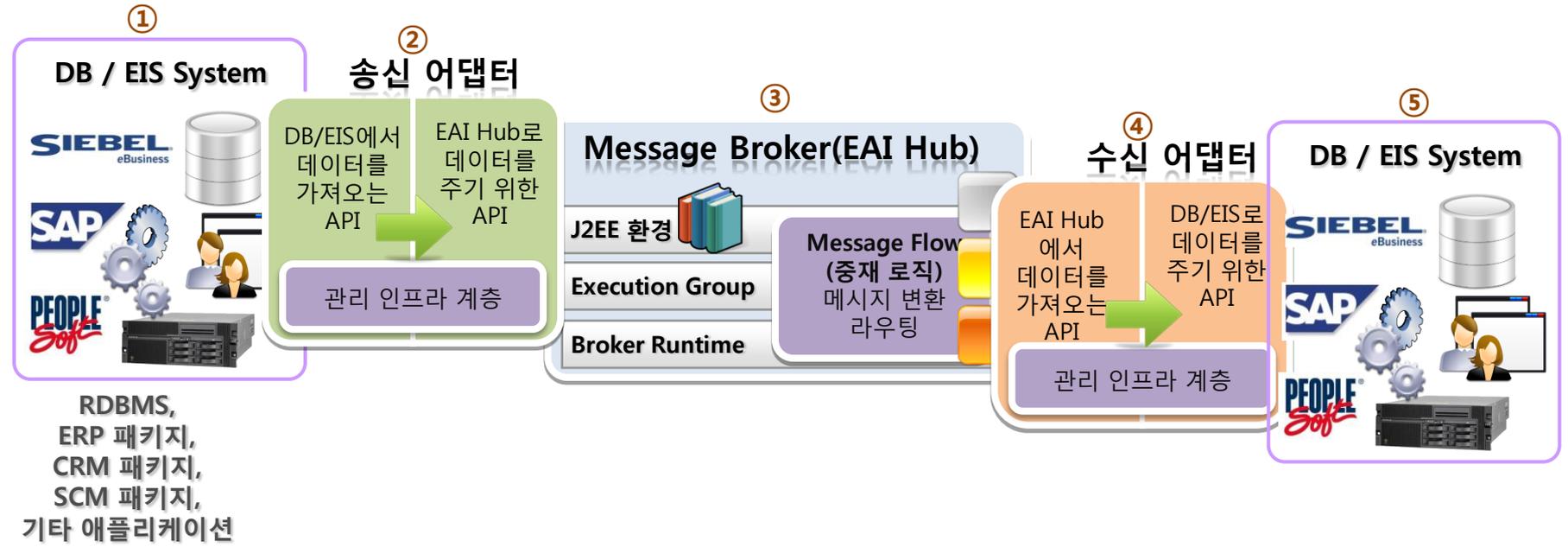
다양한 프로토콜에 대해 별도 사용자 코딩 필요 없이 자동으로 변환 가능





WebSphere Message Broker: EIS & DB 연결성

JCA(J2EE Connector Architecture) 기반의 어댑터가 내장된 독립된 실행 환경 제공
다양한 패키지 애플리케이션(SAP이나 Siebel 등)에 대해 접근 중개 로직을 구성/개발/테스트





WebSphere Message Broker: 어댑터 및 연결성

다양한 상용 어댑터를 제공하며, API 제공을 통해 어댑터 개발 가능

테크놀로지

- ◆ Adapter for e-mail
- ◆ Healthcare Data Protocols
- ◆ iSeries
- ◆ JCA
- ◆ HTTP
- ◆ Web Services
- ◆ COM
- ◆ CORBA
- ◆ Exchange
- ◆ FIX Protocol
- ◆ SWIFT
- ◆ XML
- ◆ WebSphere MQ
- ◆ WebSphere MQ Integrator
- ◆ Data Handler for XML
- ◆ Data Handler for EDI
- ◆ ACORD XML
- ◆ Enterprise JavaBean
- ◆ JDBC
- ◆ JMS
- ◆ JText
- ◆ Lotus Domino
- ◆ SWIFT
- ◆ XML

패키지 어플리케이션

- ◆ i2
- ◆ i2 Active Data Warehouse
- ◆ IndusConnect Framework
- ◆ Siebel eBusiness Applications
- ◆ MetaSolv Applications
- ◆ SAP Exchange Infrastructure
- ◆ mySAP.com
- ◆ NightFire Applications
- ◆ Oracle Applications
- ◆ Portal Infranet
- ◆ QAD MFG/PRO
- ◆ Spirent Applications
- ◆ Telcordia Applications
- ◆ WebSphere Commerce
- ◆ Centricity Gateway
- ◆ ESRI Spatial Databases
- ◆ JD Edwards OneWorld
- ◆ PeopleSoft
- ◆ Ariba Buyer
- ◆ Clarify CRM
- ◆ eMatrix
- ◆ Spirent Applications
- ◆ Maximo MEA
- ◆ Manugistics Demand & Fulfillment Magt

메인 프레임

- ◆ I2
- ◆ CICS
- ◆ IMS Transaction Manager
- ◆ ADABAS
- ◆ Adapter for VSAM
- ◆ DB2 Databases
- ◆ IMS Database Manager
- ◆ Natural
- ◆ IDMS Database

지원 언어 및 API

- ◆ C/C++/C#
- ◆ Java
- ◆ COBOL/PL1/RPG/REXX
- ◆ .NET(VB, ASP)
- ◆ JMS/JMS Realtime
- ◆ XMS
- ◆ AMI/CMI/MQI
- ◆ CMI
- ◆ MQI



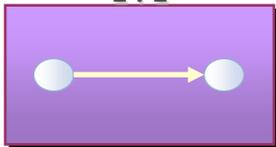
WebSphere Message Broker: 지원되는 통신 프로토콜



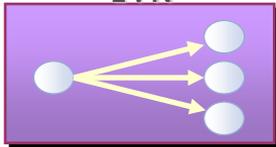
동기, 비동기, PUB/SUB 등 다양한 통신 패턴 구현

Asynchronous

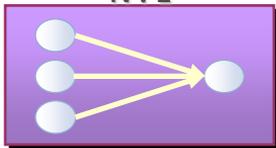
1 : 1



1 : N

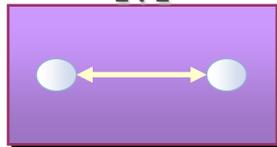


N : 1

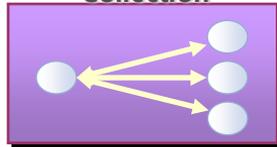


Synchronous

1 : 1



Collection

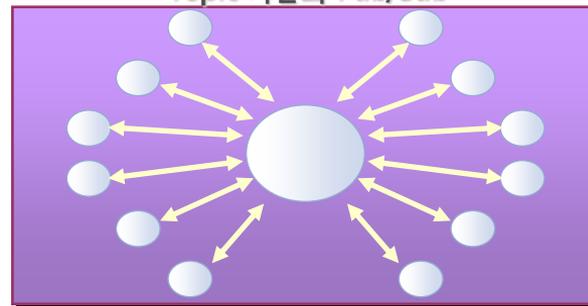


Aggregate

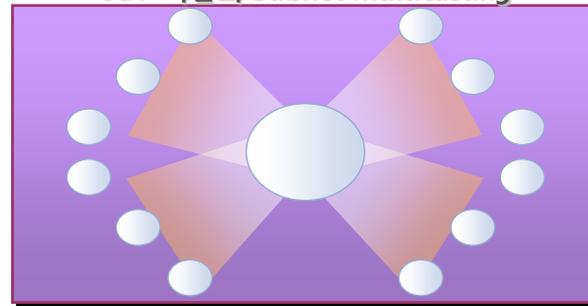


Publish & Subscribe

Topic 기반의 Pub/Sub



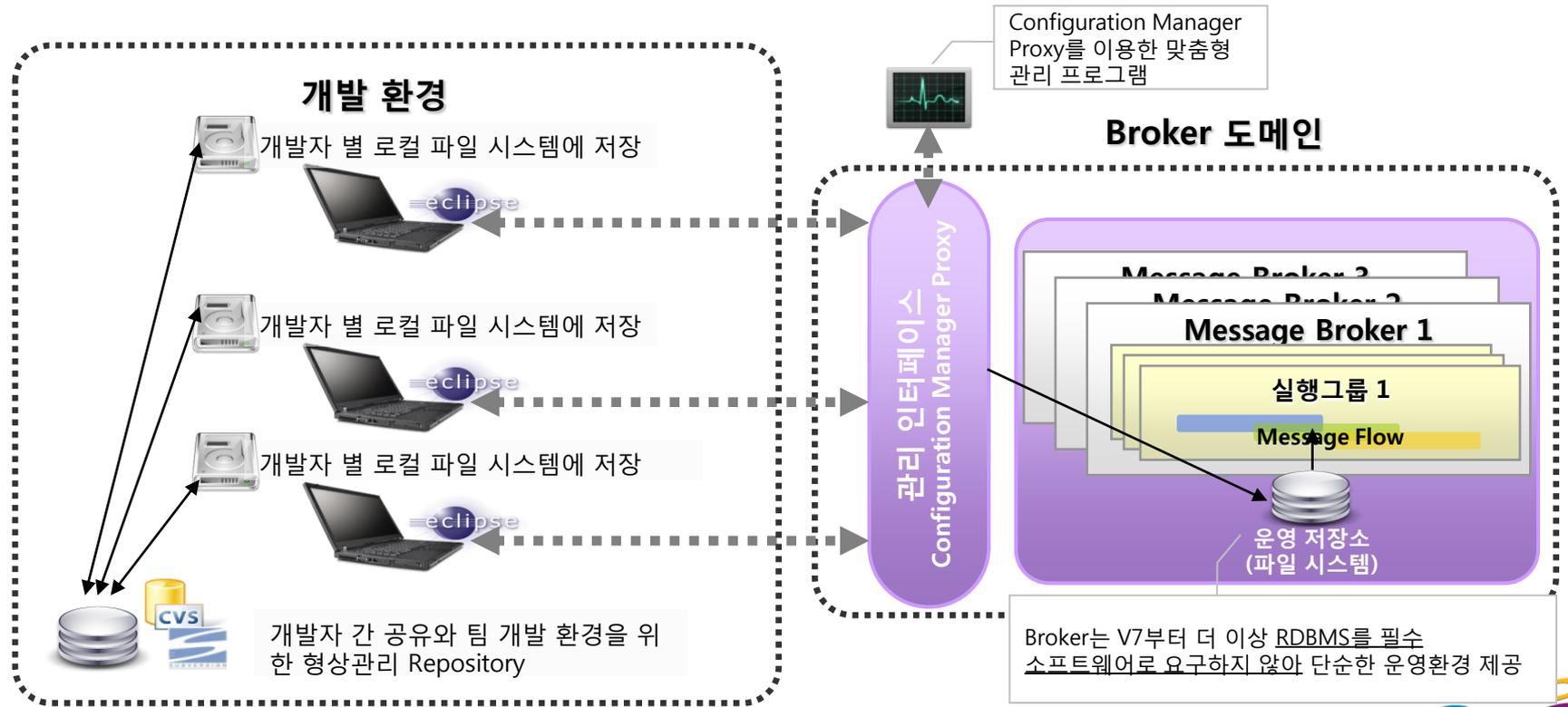
UDP 기반의 Subnet Multicasting





WebSphere Message Broker: 개발 아키텍처

개방형 표준인 Eclipse 기반의 통합 개발 환경을 제공



WebSphere Message Broker: 개발 도구 - Broker Toolkit



Message Broker Toolkit을 사용하여 EAI 솔루션과 관련된 리소스를 개발, 관리
필요한 경우 플러그인을 개발하여 추가적인 기능을 탑재할 수 있는 유연성 제공

The screenshot displays the WebSphere Message Broker Toolkit interface. The main area shows a message flow diagram with nodes: FromLegacy, StoreLegacyReqAndMsgId, ParseAsMRM, BuildSAPReq, SAP Request, BuildLegacyRep, Trace1, Trace2, Trace3, and ToLegacy. The flow starts at FromLegacy, goes to StoreLegacyReqAndMsgId, then to ParseAsMRM, which branches to BuildSAPReq and BuildLegacyRep. BuildSAPReq leads to SAP Request, which leads to BuildLegacyRep, which leads to Trace3. Trace1 and Trace2 are also connected to the flow. The flow ends at ToLegacy. The diagram is labeled "편집창 영역" (Editing Area).

On the left, there is a tree view showing the project structure, including "HMMIntegration" and "HMMLegacyMsgSet". This area is labeled "탐색창 영역" (Search Area).

At the bottom left, there is a list of nodes in the flow, including "FromLegacy", "StoreLegacyReqAndMsgId", "ParseAsMRM", "BuildSAPReq", "SAP Request", "BuildLegacyRep", "ToLegacy", "subProcessSapException3_1", and "Trace3". This area is labeled "개요창 영역" (Overview Area).

At the bottom right, the "Properties" window is open for the "ResetContentDescriptor Node Properties - ParseAsMRM". It shows various configuration options, including "Message domain", "Reset message domain", "Message set", "Reset message set", "Message type", "Reset message type", "Message format", and "Reset message format". This area is labeled "등록정보창 영역" (Registration Information Area).

The top right corner of the interface has the text "Perspective 전환" (Switch Perspective).



WebSphere Message Broker: 개발 도구 - 노드

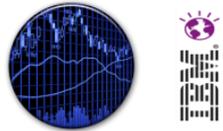


Message Broker Toolkit은 다양한 기술을 사용하는 노드를 빌트인하여 개발 편의성 제공

즐거찾기	WebSphere 어댑터	웹 서비스	JMS
WebSphere MQ	PeopleSoftInput	SOAPInput	JMSInput
JMS	PeopleSoftRequest	SOAPReply	JMSOutput
HTTP	SAPInput	SOAPRequest	JMSReply
웹 서비스	SAPRequest	SOAPAsyncRequest	JMSHeader
SCA	SAPReply	SOAPAsyncResponse	JMSMQTransform
WebSphere 어댑터	SiebelInput	SOAPEnvelope	MQJMSTransform
라우팅	SiebelRequest	SOAPExtract	
변환	JDE JDEdwardsInput	RegistryLookup	데이터베이스
구성	JDE JDEdwardsRequest	EndpointLookup	DatabaseInput
데이터베이스	TwineBallInput	HTTP	Database
파일	TwineBallRequest	HTTPInput	DataDelete
전자 우편	WebSphere MQ	HTTPReply	DataInsert
TCP/IP	MQInput	HTTPRequest	DataUpdate
CORBA	MQOutput	HTTPHeader	Warehouse
CICS	MQReply		DatabaseRetrieve
IMS	MQGet		DatabaseRoute
유효성 검증	MQHeader		
보안			
타이머			

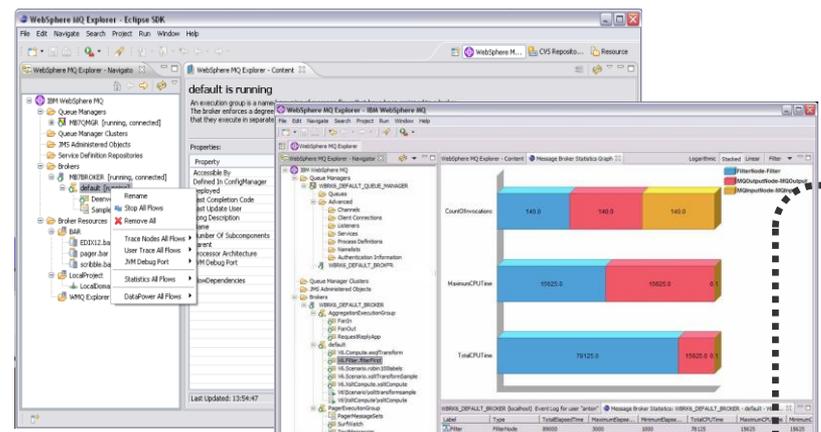
대략 100여 개의 Node가 Built-In





WebSphere Message Broker: 관리 도구 - 탐색기

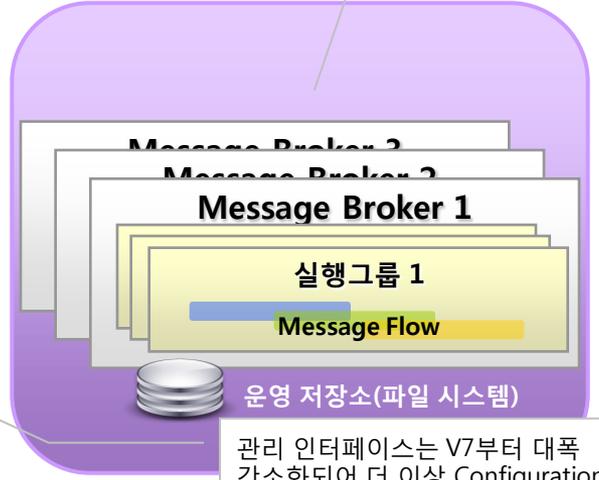
WebSphere Message Broker 탐색기를 통해 WMQ 및 WMB를 동시에 관리



실행 그룹 내부에 개개의 메시지 플로우가 배포되어 실행

Broker 도메인

관리 인터페이스
Configuration Manager Proxy



Explorer를 통해, MQ 및 MB 구성정보 확인, Broker 및 실행그룹 시작/중지와 AP 배포/삭제와 같은 작업을 수행할 수 있음.

관리 인터페이스는 V7부터 대폭 간소화되어 더 이상 Configuration Manager가 필요 없음





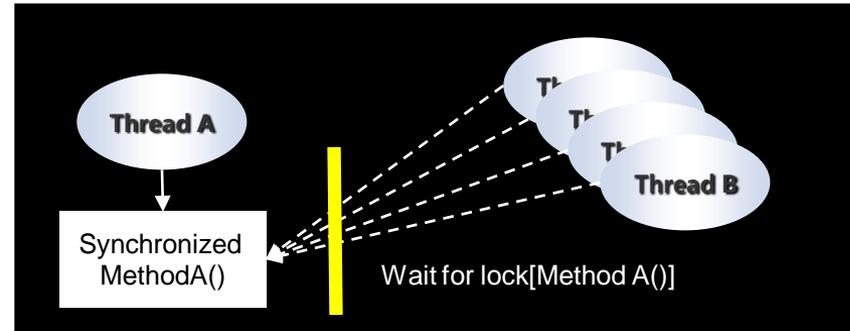
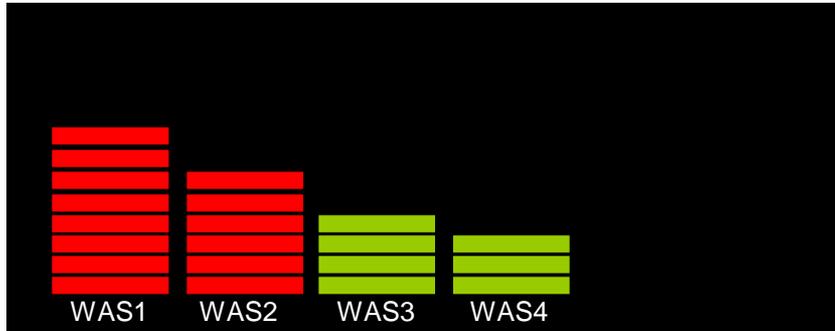
WebSphere Message Broker: J2EE EAI 솔루션과 차별성

WAS Runtime Environment

- 1 java processes + multiple threads
- JVM에 의존적(Resouce:Memory 할당 + JVM Crash)
- 완벽한 Asynchronous Call model을 지원하는가?

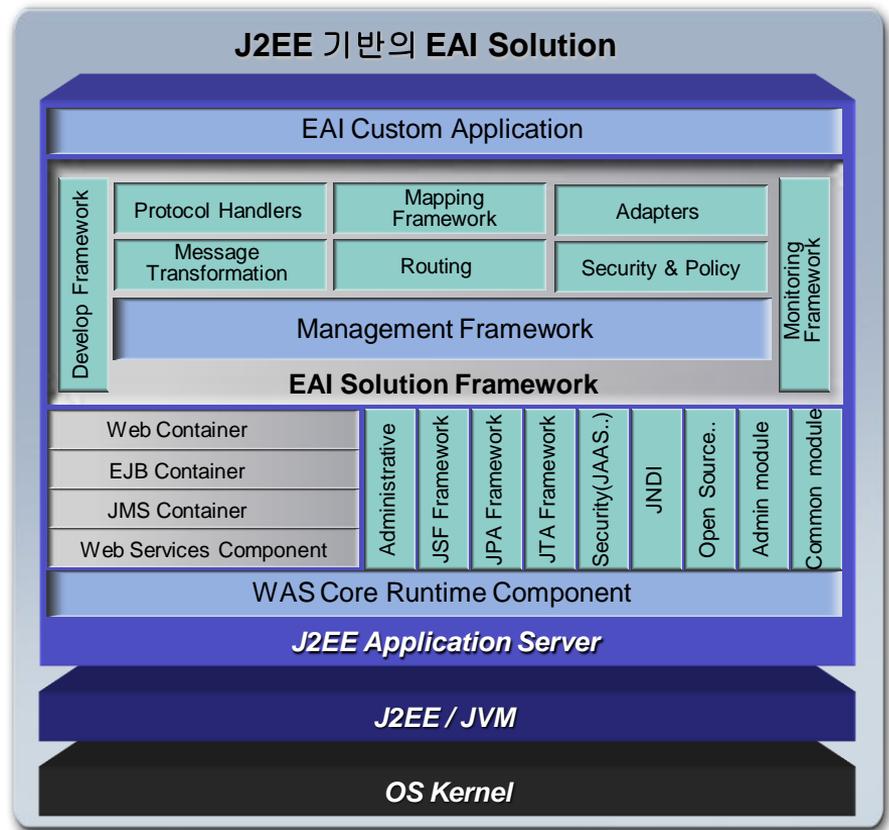
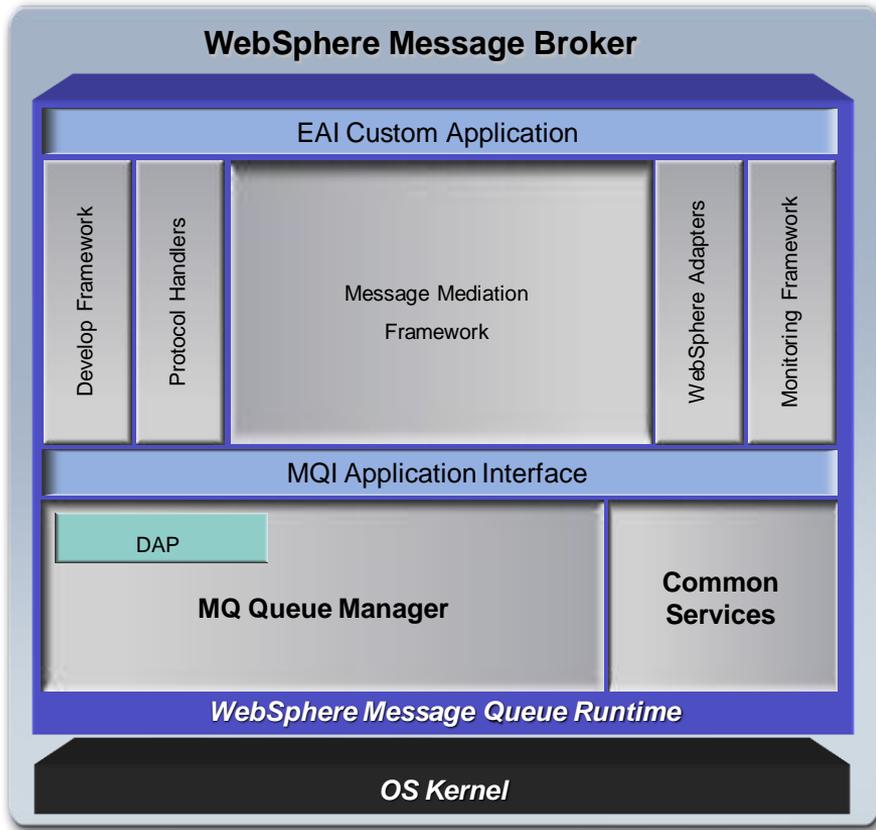
Wait for IO Response + lock Contention

- 외부 시스템(DB/Legacy) 호출 후, 응답 대기
- Synchronized block을 이용한 Framework 호출 후, Lock 점유를 위한 thread간의 경합 및 대기 현상





WebSphere Message Broker: J2EE EAI 솔루션과 차별성



개발 환경에 있어서의 차별성

Message Broker

- 표준화된 eclipse 기반의 개발 도구 제공: Message Broker Toolkit
- I/F 개발 시, Drag & Drop이외 추가 개발 최소화
- 다양한 테스트 방법 제공
- 개발된 다양한 Message Flow 패턴 제공: 개발 기간 단축의 효과.

EAI Solution based on J2EE(WAS)

- 상위 4개 Vendor 제품을 분석한 결과, 70%의 제품만이 eclipse만을 제공.
 - I/F 개발 시, Component Drag & Drop이후, 생성된 .java 소스를 확장해야 함.
- : 실제 제품 framework 및 개발자 code가 동일 소스에 존재.



전개 환경에 있어서의 차별성

Message Broker

- 단순한 EAI AP 아키텍처
- 손쉬운 전개 단계: 단일 단계 전개
- 빠른 전개 시간: 실행 그룹별로 메시지 플로우가 50이 넘지 않는 것을 권장
- 변경 사항 동적 반영: Hot Deploy

EAI Solution based on J2EE (WAS)

- 복잡한 EAI AP 아키텍처: J2EE 애플리케이션 구조(EAR 또는 JAR)로 존재해야 함
- 단계별 전개 단계: J2EE 서버에 단계별 전개 절차 필요 (WAS에 종속적)
- 변경 사항 동적 반영 불가능: Hot Deploy가 제한적으로 적용되기 때문에, 서버 인스턴스를 재시작해야 하는 불편함이 존재



성능에 있어서의 차별성



Message Broker

- 독립된 실행 환경: I/F별로 개발된 메시지 플로우가 독립된 실행 그룹에서 실행 가능
- 완벽한 동기/비동기 Call Model 제공
- 대용량 데이터 적재 가능
예) 80,000건-2GB Heap memory 할당: 30초 소요

EAI Solution based on J2EE(WAS)

- 종속적 실행 환경: 개발된 AP는 EAR로 존재하기 때문에, I/F간 의존성 및 의존적 실행 환경
- JVM에 의존적인 성능: Multi-Threads Runtime 환경, Lock Contention, Resource Allocation, GC
예) 80,000건- 6GB Heap memory 할당: 9분 30초 소요
- 동기적 Call Model 제공: WAS의 기본 사상인 Synchronous call Model (Request & Response)
JMS를 통해 Asynchronous Call Model을 지원한다고 하나, 이게 완전한 것인가?





문제 확인에 있어서의 차별성



Message Broker

- 독립된 EAI 로깅 정책: EAI Trace가 인프라 제품인 MQ와 별도로 존재
- 문제점 식별 용이
- 동적 로깅 정책 변경 가능

EAI Solution based on J2EE(WAS)

- 일반적으로, EAI 솔루션 자체 로그 파일 및 인프라 제품인 WAS의 로그 파일로 존재: 문제 발생 시, 로그 분석 어려움
- 문제점 식별 어려움: 사용자 로직 + EAI 솔루션 프레임워크가 혼용되어, 문제점 분석이 어려움
- 동적 로깅 정책 변경 불가능: 문제 확인을 위해 Trace level 조정 시, WAS를 재시작해야 하는 경우가 존재





감사합니다

