

애플리케이션 통합 영역의 기존 서비스 투자 보호를 통한 ROI 극대화

채윤정 실장, 전문위원

한국IBM 소프트웨어 그룹 웹스피어 사업부

Impact Korea 2011

Changing the Way Business and IT Leaders Work



Agenda

- SOA 솔루션 동향
- SOA 프로젝트 예제
- 결론





SOA 솔루션 동향





SOA는 기존 방식과 무엇이 다른가?



재사용성

- SOA 서비스는 기존 IT 자산을 활용하여 **재사용을 극대화**

표준

- 산업 표준을 사용하는 웹서비스의 광범위한 채택

비즈니스와 IT의 공동 참여

- SOA는 설계 과정에 현업이 참여함으로써 **비즈니스와 IT의 공동 참여(Business/IT Alignment)** 향상

집중도(Degree of Focus)

- SOA는 **비즈니스 수준**의 활동(Activity)에 초점

연결(Connection)

- SOA는 동적으로 유연하게 **서비스가 연결**



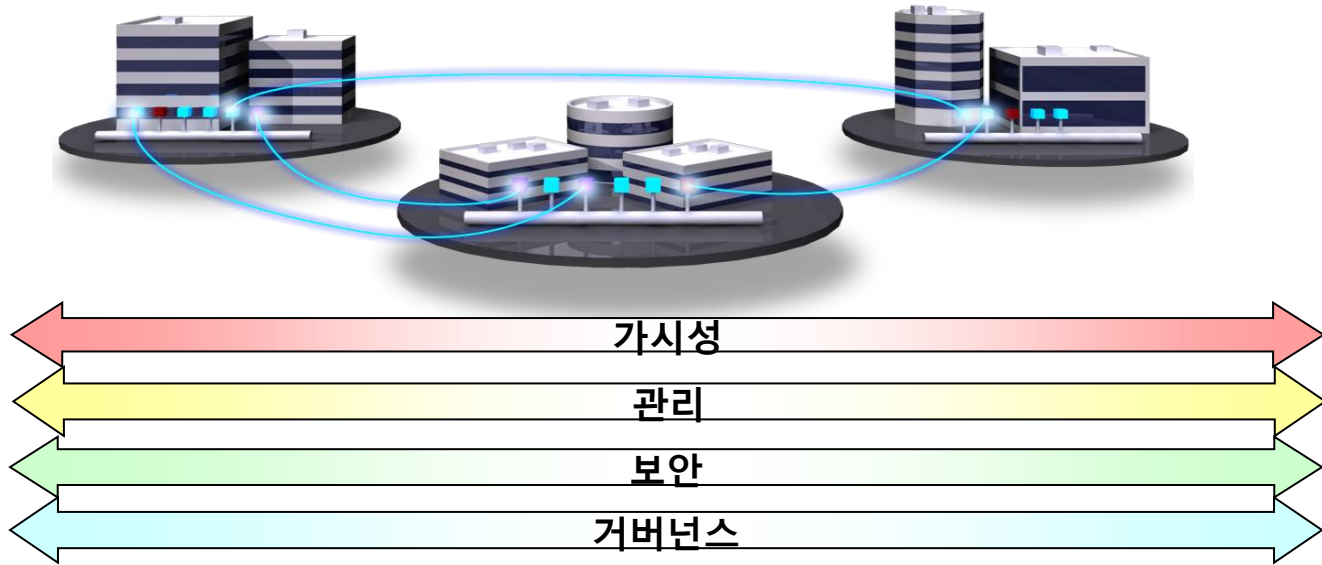
SOA 구축 경험을 통한 교훈 - SOA 솔루션 측면



- **한 가지 제품으로 모든 문제를 해결할 수 없음**
다양한 여러 제품의 ESB를 구축하고 관리하기 위한 솔루션 필요
- **구현 편의성이 중요함**
수년 간의 구축 경험에 의해 도출된 공통 패턴에 대한 지원이 필요
- **동적인 변경 요구사항**
정책에 의해 제어할 수 있는 다양한 동적인 중개 기능 필요
- **거버넌스를 위한 인프라 필요**
서비스 및 SOA 관련 관련 자산을 공유하고 라이프사이클을 관리할 수 있는 시스템 필요



서비스 연합 관리 (Service Federation Management)



- 서비스 연합 관리(Service Federation Management)는 도메인 경계에 걸쳐 서비스 재사용 가능
- 전사 규모의 서비스 가시성, 보안, 관리, 거버넌스를 위해 개별 서비스 도메인의 연결(Connectivity) 인프라 조정



서비스 연합 관리 (Service Federation Management)



The screenshot shows the Service Federation Management interface. On the left, a diagram illustrates service connections: eBusiness and Warehouse are connected to Retail, and Retail is connected to Sales. On the right, the 'Share Editor' window displays a table of shares with red 'X' marks indicating errors.

Source group	Target group	
storeServices	storeServices @ Retail	X
customerServices	customerServices @ Retail	X
orderServices @ Warehouse	orderServices @ Warehouse @ Retail	X

WSRR (WebSphere Service Registry & Repository)

• 연합 모델(Federation Model)

연합된 콘텐츠에 대한 단일 보기 제공

• 프로토콜

서비스 연결 및 각 도메인을 지원하는 등록소(Registry)에 연결하기 위한 단순한 서비스 관리 프로토콜 (Web 2.0)

• 콘솔

멀티 벤더의 런타임 환경을 관리하기 위한 연합 콘솔

서비스 연합 관리(Service Federation Management)는 연결(Connectivity) 인프라를 연결하는 측면보다는 도메인 간 서비스 공유 극대화에 초점

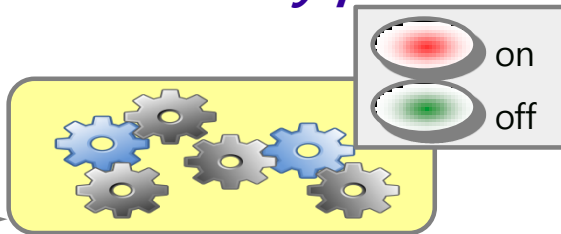


패턴 기반 개발

Evolution of approach to connectivity problems



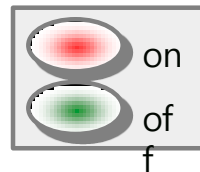
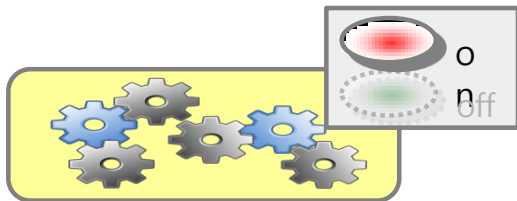
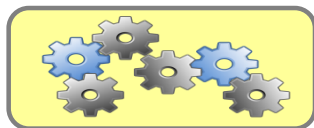
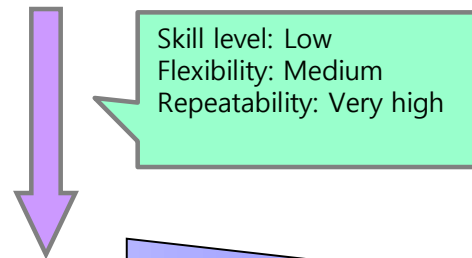
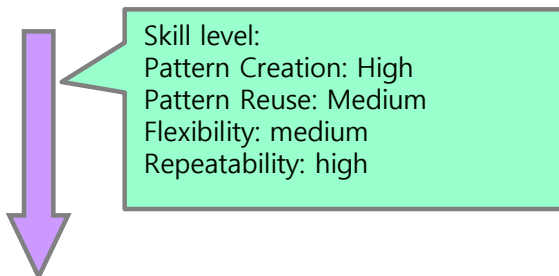
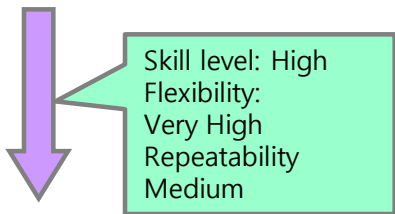
Toolkit with Micro-Patterns



Macro-Patterns



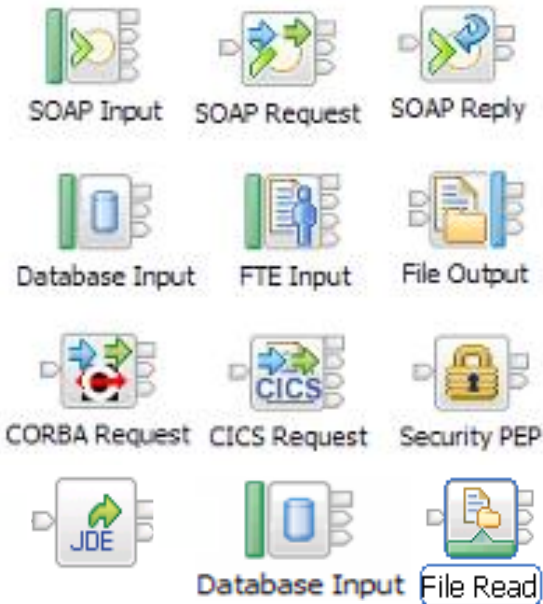
Mega-Patterns



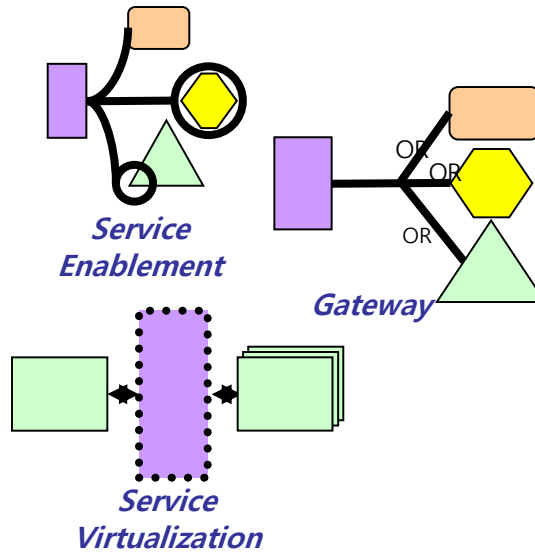
패턴 기반 개발



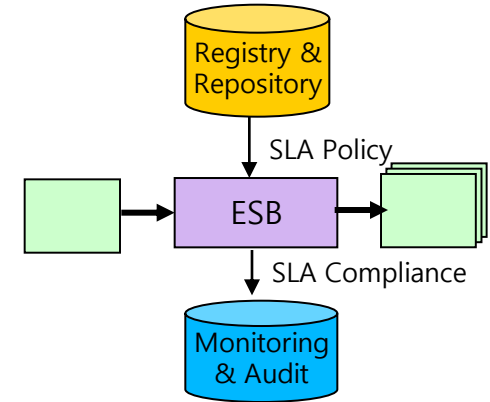
Micro Pattern



Macro Pattern



Mega Pattern



www.ibm.com/developerworks/wiki/display/esbpatterns/



패턴 기반 개발 - 예시 (Macro Pattern)



The screenshot displays the IBM WebSphere Message Broker Toolkit interface. The left pane shows a tree view of patterns, with 'Message Correlator' and 'MQ request-response with persistence' highlighted. The main pane shows the 'View Pattern Specification' for the selected pattern. The specification includes a title, a description, and a flow diagram. The flow diagram illustrates the process: three 'Requesting Application' boxes send requests to a 'Message Correlator for WebSphere MQ: request-response with persistence' box. This box contains steps: 'Store request message header', 'Set new reply queue', and 'Retrieve request message header'. The 'Set new reply queue' step connects to a 'Provider Application' box. The 'Retrieve request message header' step connects to a 'Set original reply queue' step, which then sends responses back to the 'Requesting Application' boxes. A 'System queue' is also shown as part of the process.

Message Correlator for WebSphere MQ: request-response with persistence pattern

Use the Message Correlator for WebSphere MQ: request-response with persistence pattern to accept requests from many client applications on a single queue, and to return responses to the correct client by using transactional flows and persistent WebSphere MQ messages.

Because this pattern uses reliable communications with persistent messages and transactional flows, it is appropriate to use when the message interchanges result in updates that require no loss of data.

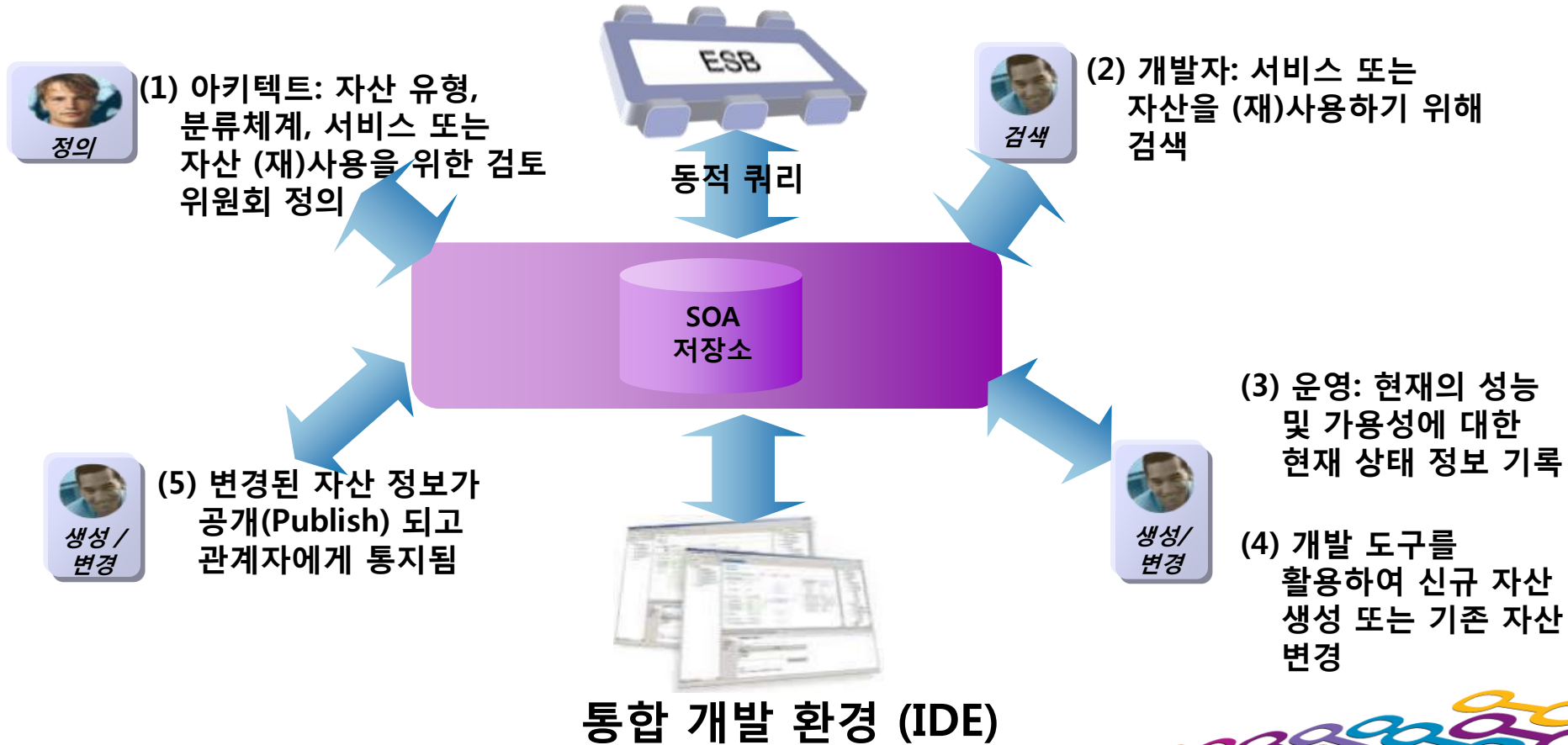
Solution

Create New Instance

IBM
WebSphere
Message
Broker Toolkit

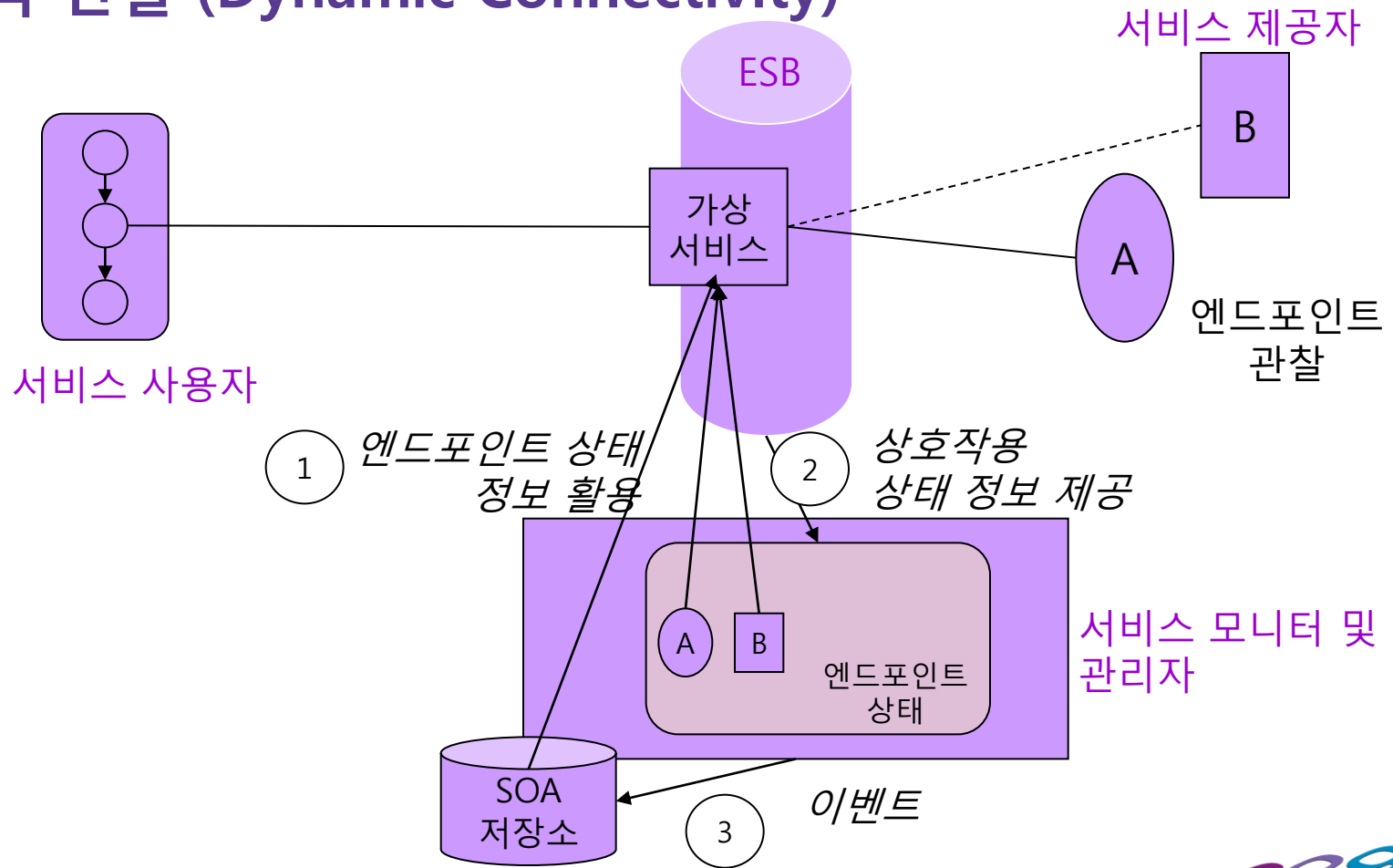


통합된 서비스 라이프사이클 관리

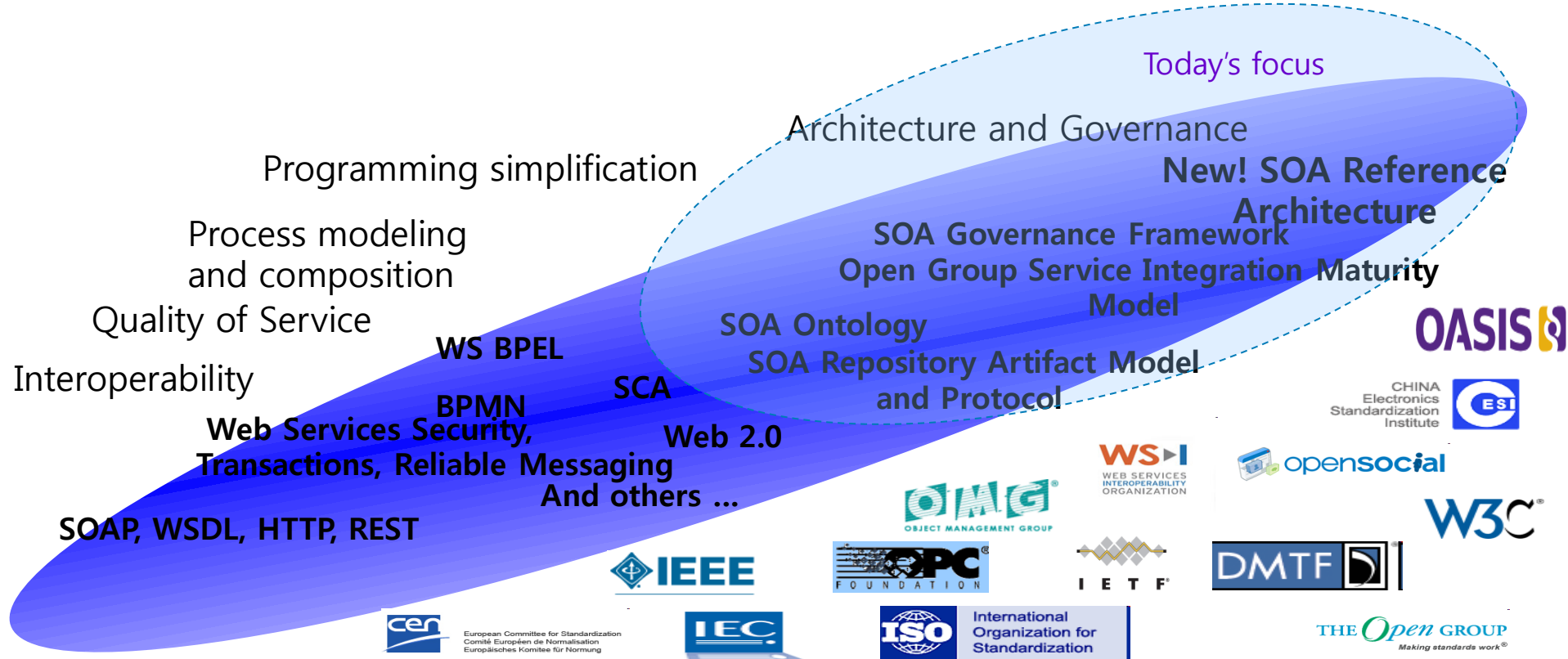




동적 연결 (Dynamic Connectivity)



SOA 솔루션 관련 표준





SOA 적용 사례 – DDB Group





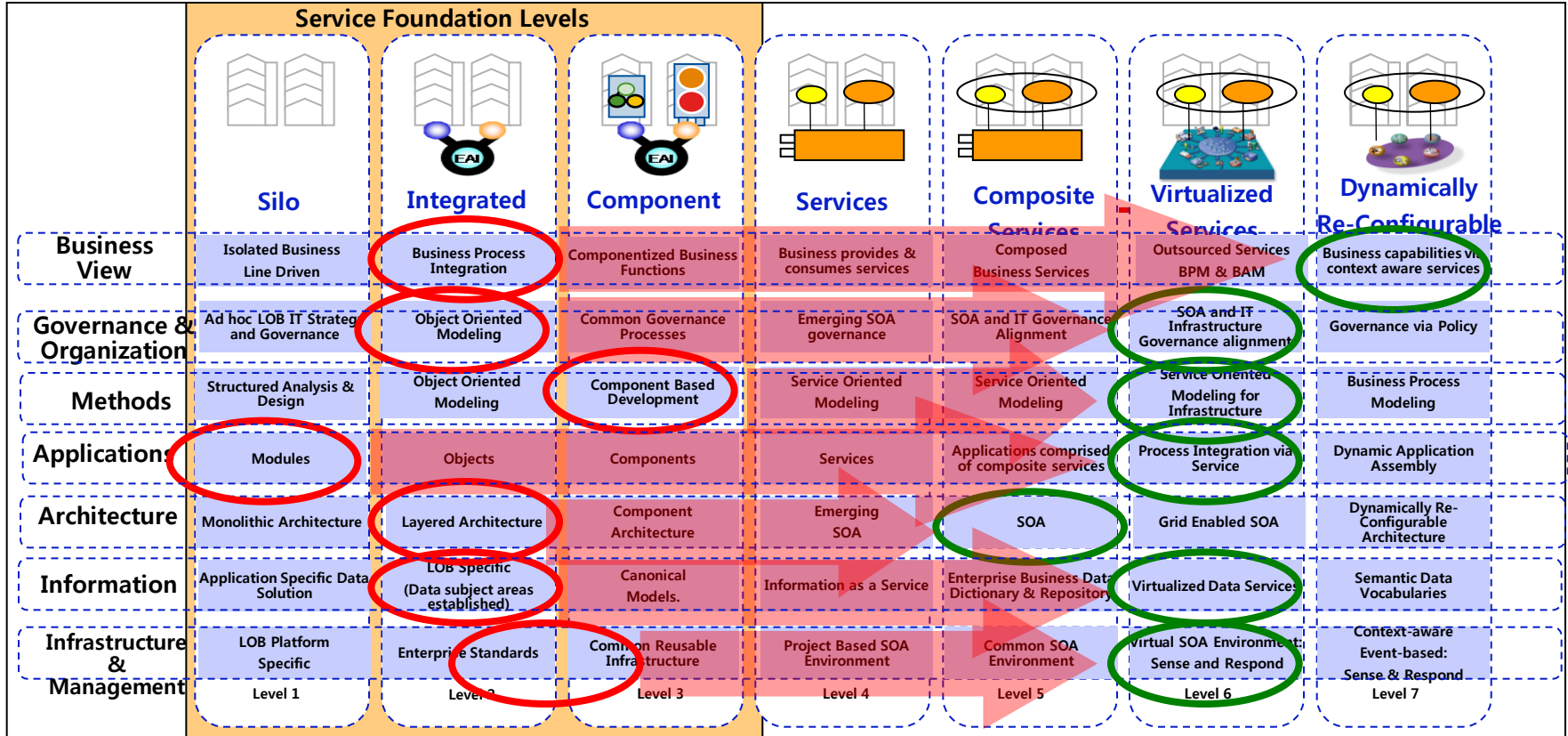
DDB Group의 SOA 여정 – 성공을 위한 5단계

- 현재의 역량을 이해하고 나서 SOA 로드맵 정의:
 1. DDB Group의 SOA 성숙도 정의 (OSIMM)
 2. SOA 적용 대상 프로세스 선정 (예: 운송관리)
 3. 참조 아키텍처를 활용하여 SOA 솔루션 정의 (SOA RA)
 4. DDB Group을 위한 거버넌스 체계 정의 (SOA Governance Framework)
 5. SOA Ontology 및 S-RAMP를 활용
 - * S-RAMP: SOA Repository Artifact Model and Protocol



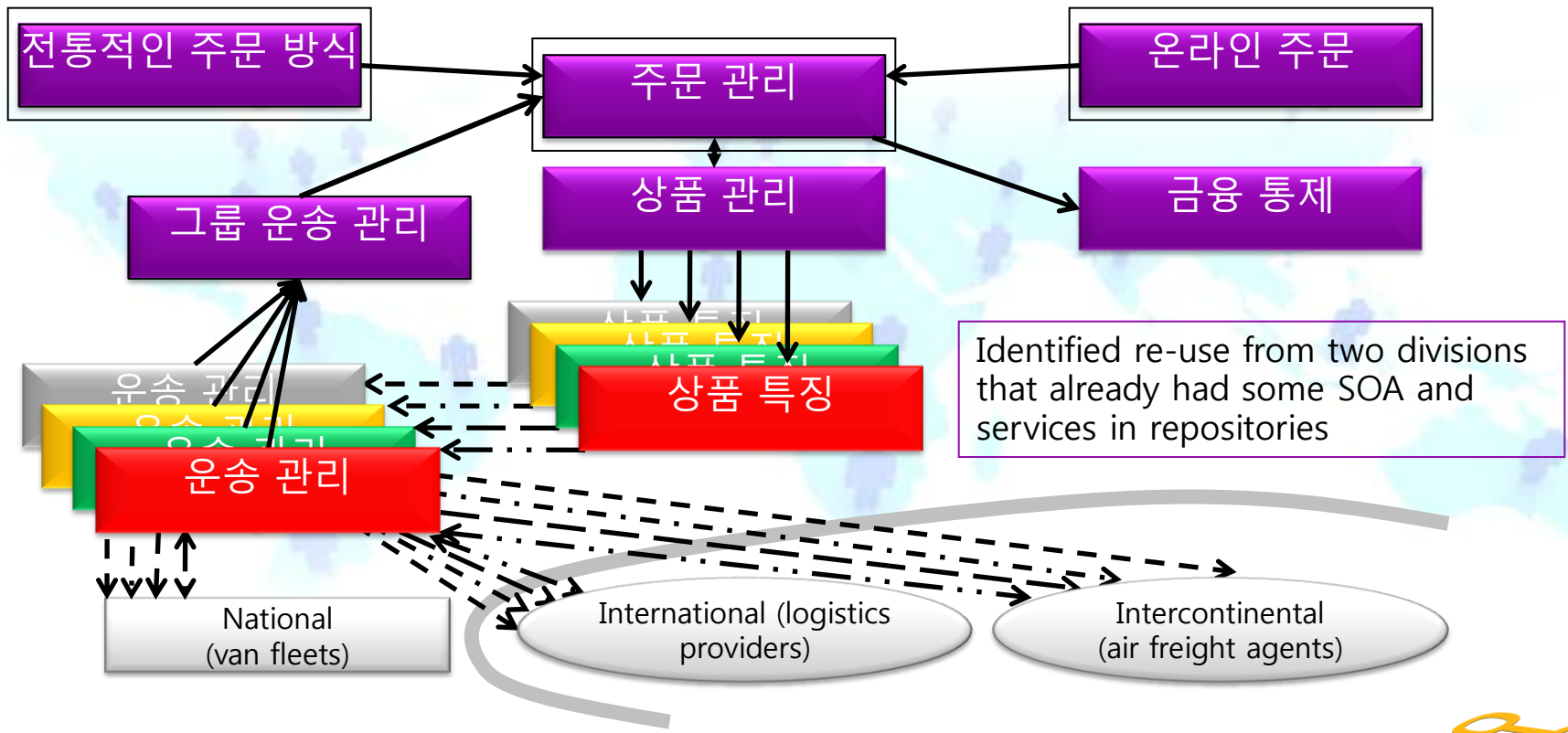
Step 1 – DDB Group의 SOA 성숙도 정의

신속하게 SOA 로드맵을 정의하기 위해 OSIMM 활용



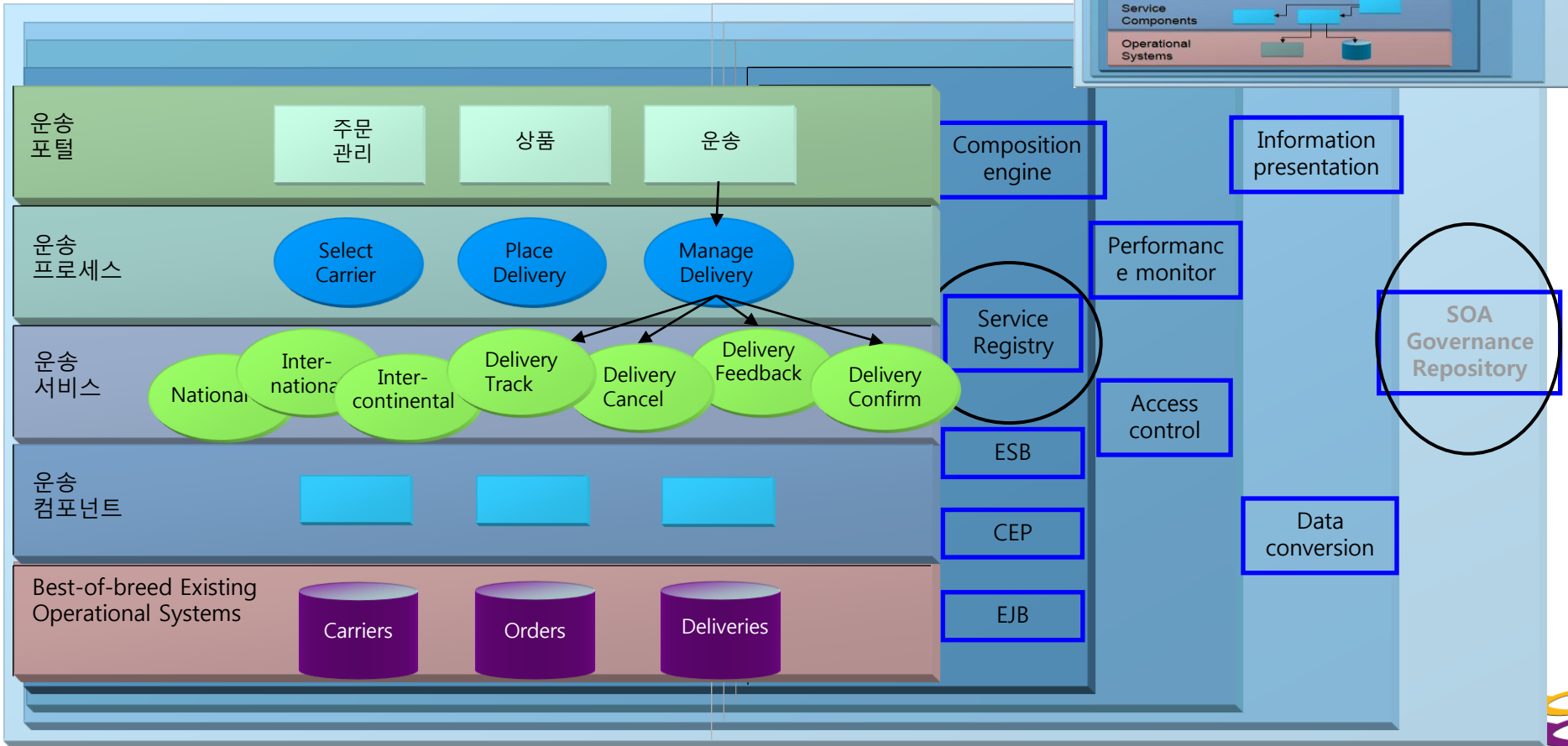
Step 2 – SOA 적용 대상 프로세스 선정

운송관리 프로세스



Step 3 – SOA Reference Architecture

솔루션 아키텍처를 정의하기 위해 SOA RA 활용



Step 4 – SOA 거버넌스 프레임워크

SOA 거버넌스 체계를 위한 목표 수립



– 사람

- 조직 변경 (거버넌스 위원회 추가 등)

– 프로세스

- SOA 솔루션 및 서비스 라이프사이클, 포트폴리오 프로세스

– 기술

- 등록 저장소
- 인프라
- 관리 및 모니터링

– 거버넌스 이행 계획





Step 4 – SOA 거버넌스 프레임워크

보다 나은 ROI를 위해 재사용 활성화를 위한 거버넌스 프로세스 정의

Compliance

Dispensatin

Communication

- 가용한 서비스에 대한 등록소(Registry) 유지
- 개발자가 요건을 충족하는 기존 서비스가 있는지 점검
- 신규 서비스가 도출되면 등록소(Registry)에 '신규 서비스 요청'을 제출
- 거버넌스 위원회는 서비스 오너를 위해 펀딩(Funding)에 대한 산출 작업
- 신규 서비스 개발 전에 거버넌스 위원회로부터 승인
- 신규 서비스에 대한 레지스트리 레코드의 uniqueService 속성 등록소(Registry)에 위원회의 승인 완료 상태 설정

- 서비스 재개발 예외에 대한 거버넌스 위원회 신청

- 개발자에게 신규 서비스에 대한 정보 제공
- 신규 서비스에 대한 검토완료(signoff) 요구사항

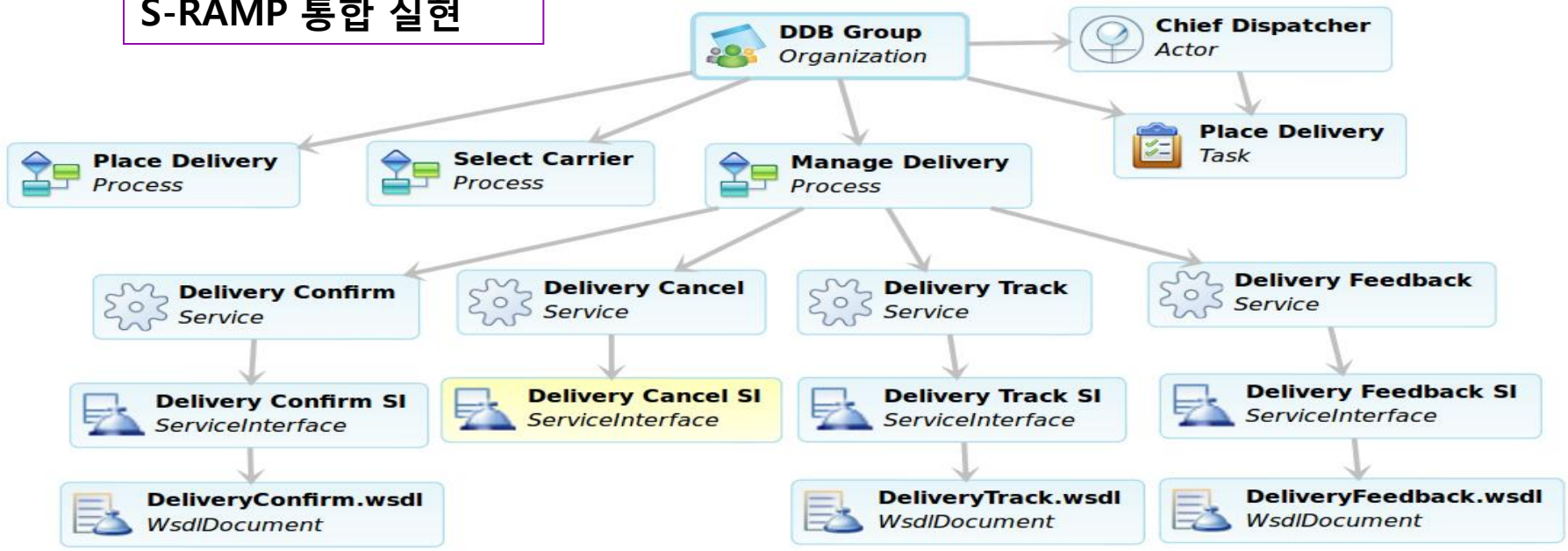




Step 5 – SOA Ontology & S-RAMP

S-RAMP를 활용하여 SOA Ontology로 서비스 정의

SOA Ontology와 S-RAMP 통합 실현



ROI를 향상시키기 위해 거버넌스 프로세스를 관리하고 정제

SOA Governance Vitality Method

정책에 대한 예외가 10% 이상 발생하면 그 정책은 다시 검토 필요 (예:uniqueness)

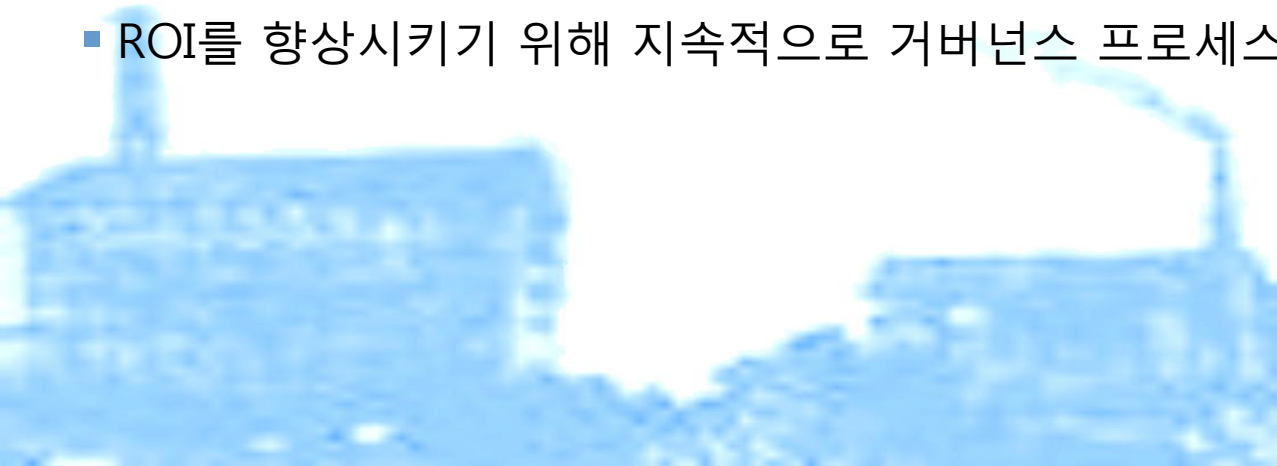
DDB Group은 등록소(Registry)에 있는 모든 서비스에 대해 uniqueService 속성을 갖도록 모니터링 수행





DDB 사례 요약

- SOA 표준 참조 모델을 활용한 아키텍처 및 거버넌스 수립:
OSIMM, SOA Governance Framework, SOA Reference Architecture 활용
- 서비스 모델과 거버넌스 인프라 구현을 위해 SOA 표준 활용
S-RAMP, SOA Ontology
- Repository를 활용하여 거버넌스 구현
- ROI를 향상시키기 위해 지속적으로 거버넌스 프로세스를 관리하고 정제





결론



결론



- 좀 더 쉽고 단순하게 : 패턴 기반의 개발
- 멀티 벤더 제품 간의 통합 (개발, 관리, 거버넌스 측면)
- 거버넌스 및 아키텍처를 위한 SOA 표준 활용
- 재활용의 극대화는 좋은 거버넌스를 통해 실현
- SOA Repository의 필요성 증대
 - 연합 서비스 관리, 동적 연결(Dynamic Connectivity) 구현, 서비스 및 SOA 관련 산출물(Artifact)에 대한 라이프 사이클 관리





감사합니다

