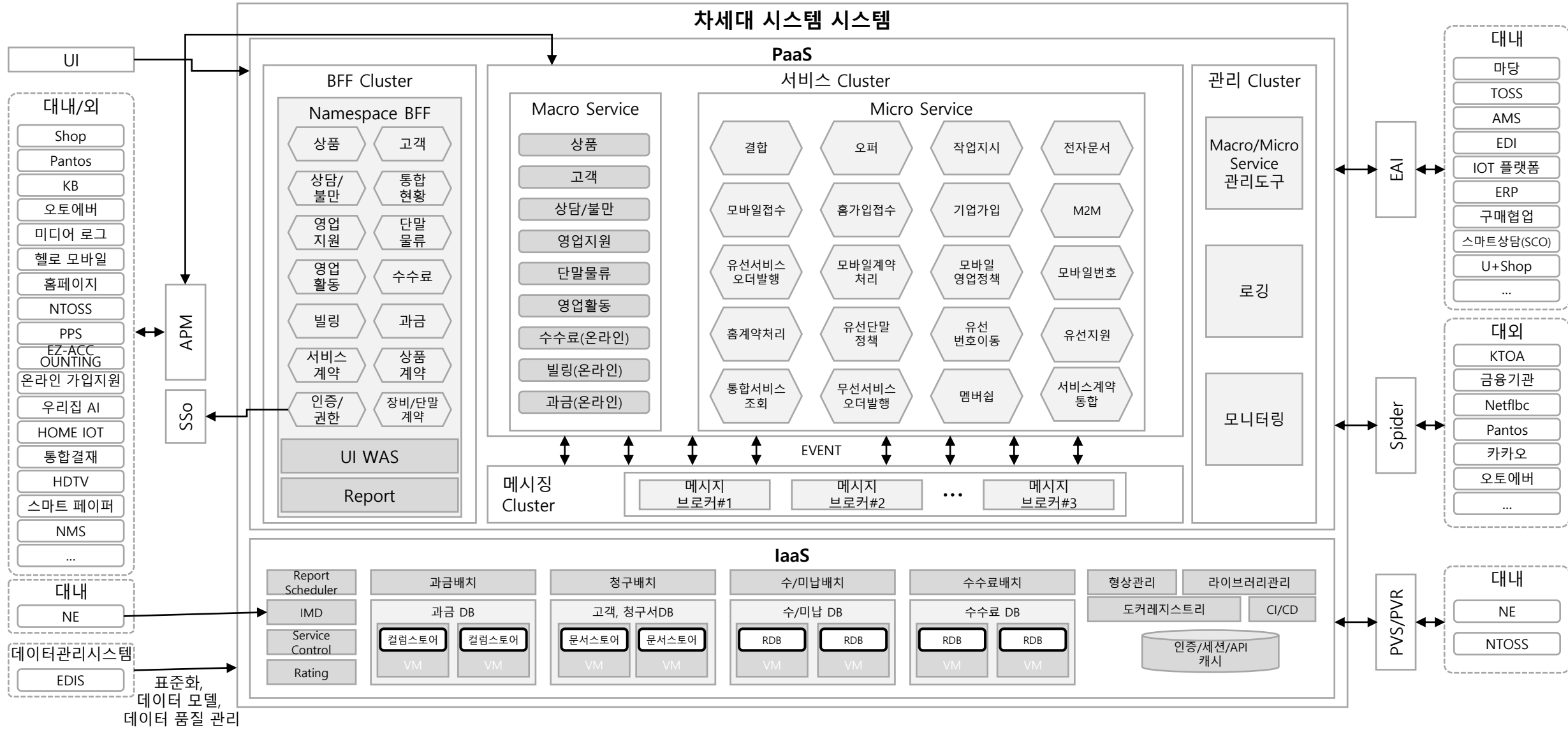
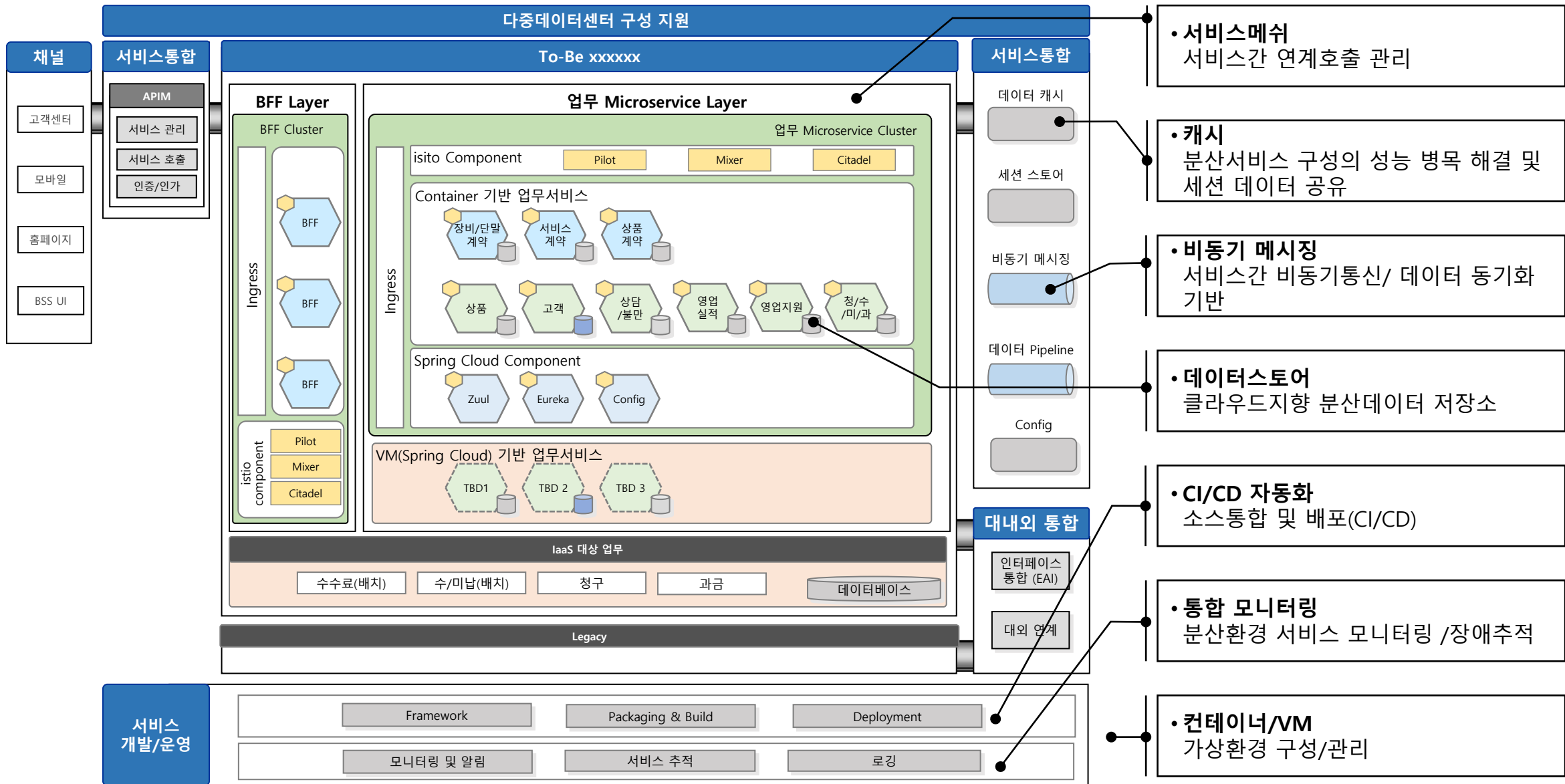


차세대 시스템 아키텍처

시스템 구성도(총 48개 서비스와 48개의 DB로 구성)



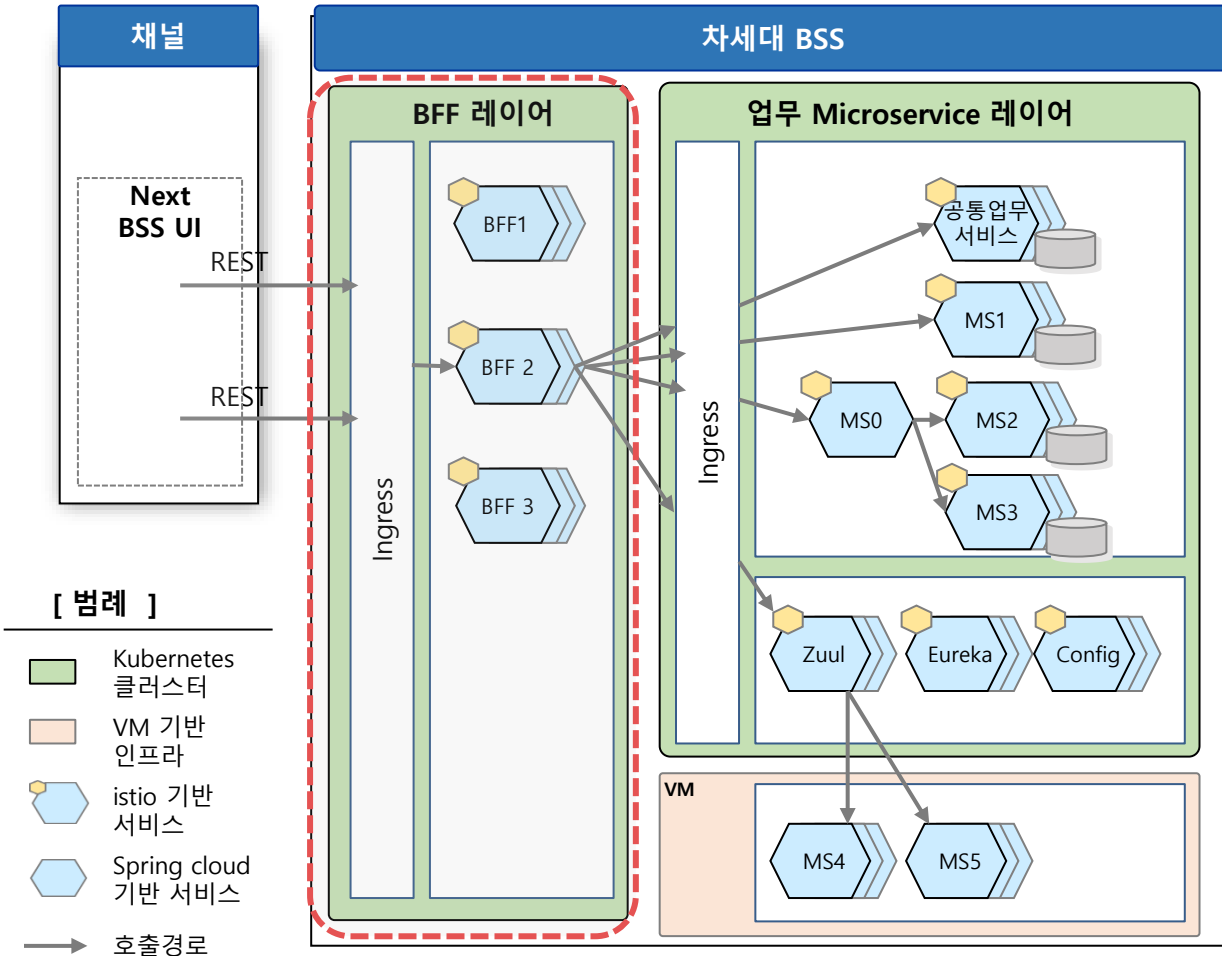
별첨 - 차세대 시스템 TO_BE 이미지 & 요소기술



별첨 – 서비스 메쉬 BFF 아키텍처

XX XX의 차세대 BSS MSA는 비즈니스 로직에 독립적인 UI 대응을 위한 BFF 레이어와 비즈니스 로직을 담당하는 업무 Microservice 레이어로 구성됨

>To-Be BFF Layer 및 BFF 구성 아키텍처



BFF 정의 및 레이어 구성

- BFF 정의**
 - Backend For Frontend의 약어로, Frontend 지원을 위한 Microservice들의 집합
- BFF 레이어 구성기준**
 - 화면처리 항목과 디바이스 유형에 따라 분리(예: PC용 화면과 모바일 화면)
 - 메뉴의 계층 구조를 기반으로 식별¹⁾

SAMPLE

BFF 역할

- 화면 UI 서비스 요청처리 및 분산된 Backend 서비스와 interaction:** 화면 고유의 서비스 처리 요청을 담당하고, Backend 서비스와 Interact.
- Composition:** 화면에서 조회 요청 시 기존 규모가 큰 서비스를 작은 Microservice로 분산된 것을 순차 처리 및 결과를 조립하여 화면 UI에 전달.
- 라우팅:** 화면UI 요청 서비스가 VM기반 분산 어플리케이션인 서비스인 경우 라우팅(분기) 기능.
- SSO 연동 :** SSO 서비스를 호출하여 ID 체크로 1차 인증
- 파일접근:** UI에서 파일 Upload /Download시 파일 접근
- 세션 스토어 :** 사용자정보 저장/변경

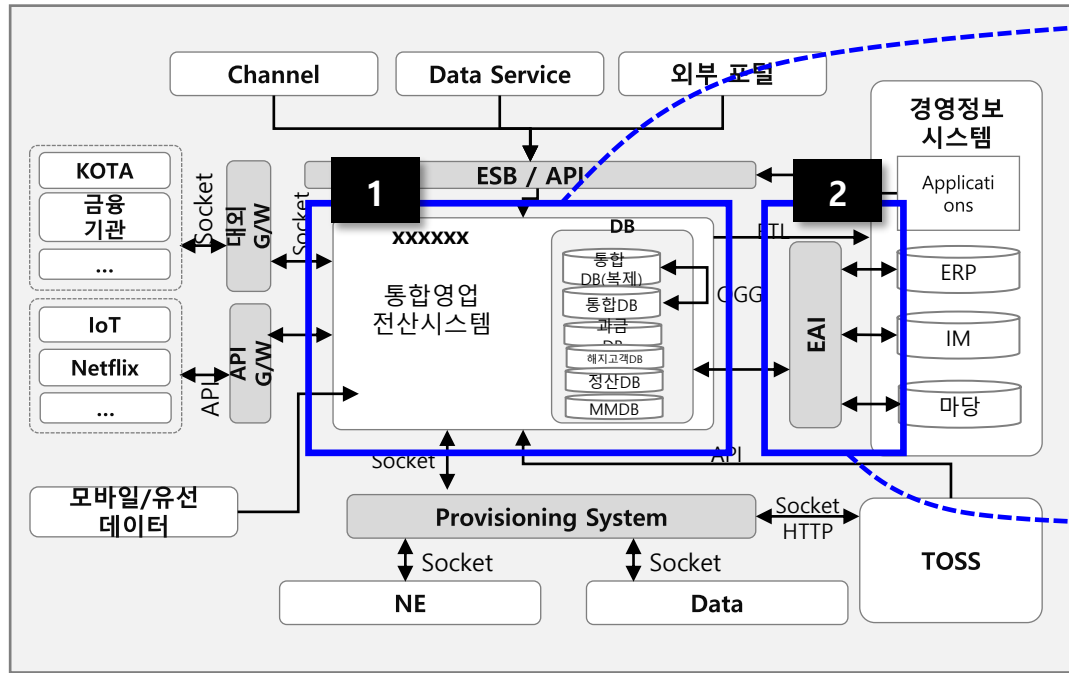
주1) BFF Microservice는 업무 Microservice와는 달리 메뉴의 계층 구조를 기반으로 식별하는 것이 일반적이므로, 향후 차세대 xxxxxx를 위한 메뉴 구조 정의 과정에서 개발/운영 조직의 R&R을 고려하여 BFF Microservice 식별 필요

별첨 – PoC 대상 솔루션

| | 구분 | | 설명 | PoC 수행 | | Reference 구축 | | 설치 | |
|----------|---------------|---------------|---------------------------|------------|-----|--------------|--------|-----|------|
| | | | | 1차 | 2차 | 설치 가이드 | 운영 가이드 | VM | PaaS |
| Data | Message Queue | | Kafka | O | O | O | O | O | O |
| | | | RabbitMQ | O | O | O | O | O | O |
| | Cache | | Redis | O | O | O | O | O | O |
| | | | Memcached | O | O | N/A | N/A | O | O |
| | DB | | MongoDB | O | O | O | O | O | N/A |
| | | | Cassandra | O | O | O | O | O | N/A |
| Platform | Service GW | | Zuul | O | O | O | O | O | O |
| | | | Ingress | O | O | O | O | N/A | O |
| | Service Mesh | | Ribbon / Eureka / Hystrix | O | O | O | O | O | O |
| | | | Istio | O | O | O | O | N/A | O |
| | Config Server | | Spring Cloud Config | O | O | O | O | O | O |
| | API GW | | IBM API Connect | N/A | N/A | N/A | N/A | N/A | N/A |
| DevOps | CI/CD | SCM | Gitlab CE | N/A | N/A | O | O | O | N/A |
| | | CD / Pipeline | Jenkins | N/A | N/A | O | O | O | N/A |
| | | | Spinnaker | N/A | N/A | O | O | N/A | O |
| | | Build Tool | Maven | N/A | N/A | O | O | O | N/A |
| | | | Gradle | N/A | N/A | O | O | O | N/A |
| | | Monitoring | | Prometheus | N/A | N/A | O | O | N/A |
| | | | Grafana | N/A | N/A | O | O | N/A | O |
| | Logging | | EFK | N/A | N/A | O | O | N/A | O |
| | Tracing | | Zipkin | N/A | O | O | O | N/A | O |

3.3 차세대 인프라 기술 – 비동기 메시징

❑ AS-IS (Monolithic)

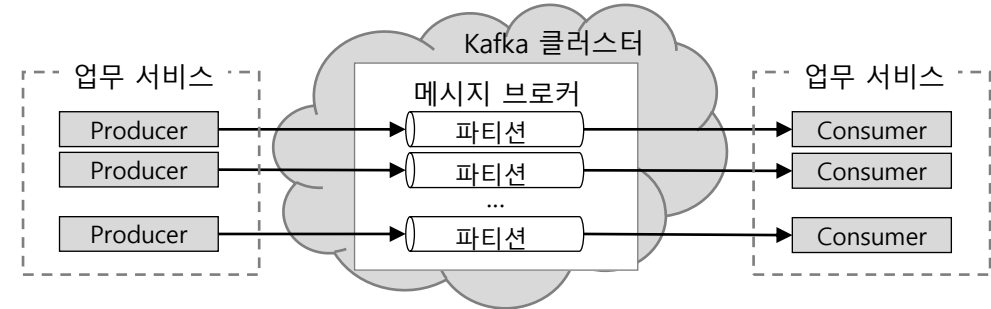


- Java / TP 기반 Monolithic 아키텍처
- 업무간 서비스 호출 기반의 동기 연계
- DB2DB 중심의 EAI 연계
- iMDB 로그 데이터 취합에 Kafka 제한 적용

- Legend -
■ 주요 인터페이스

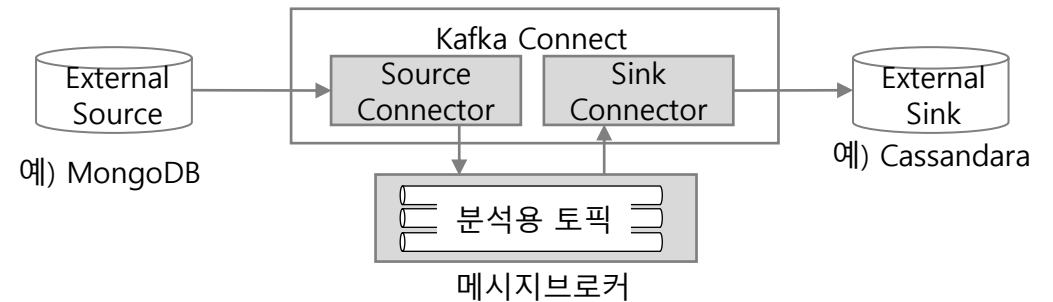
❑ TO-BE (MSA 기반 분산 전환)

1 서비스간 비동기 연계



- 업무 서비스간 Kafka 기반 비동기 메시징 사용한 연계

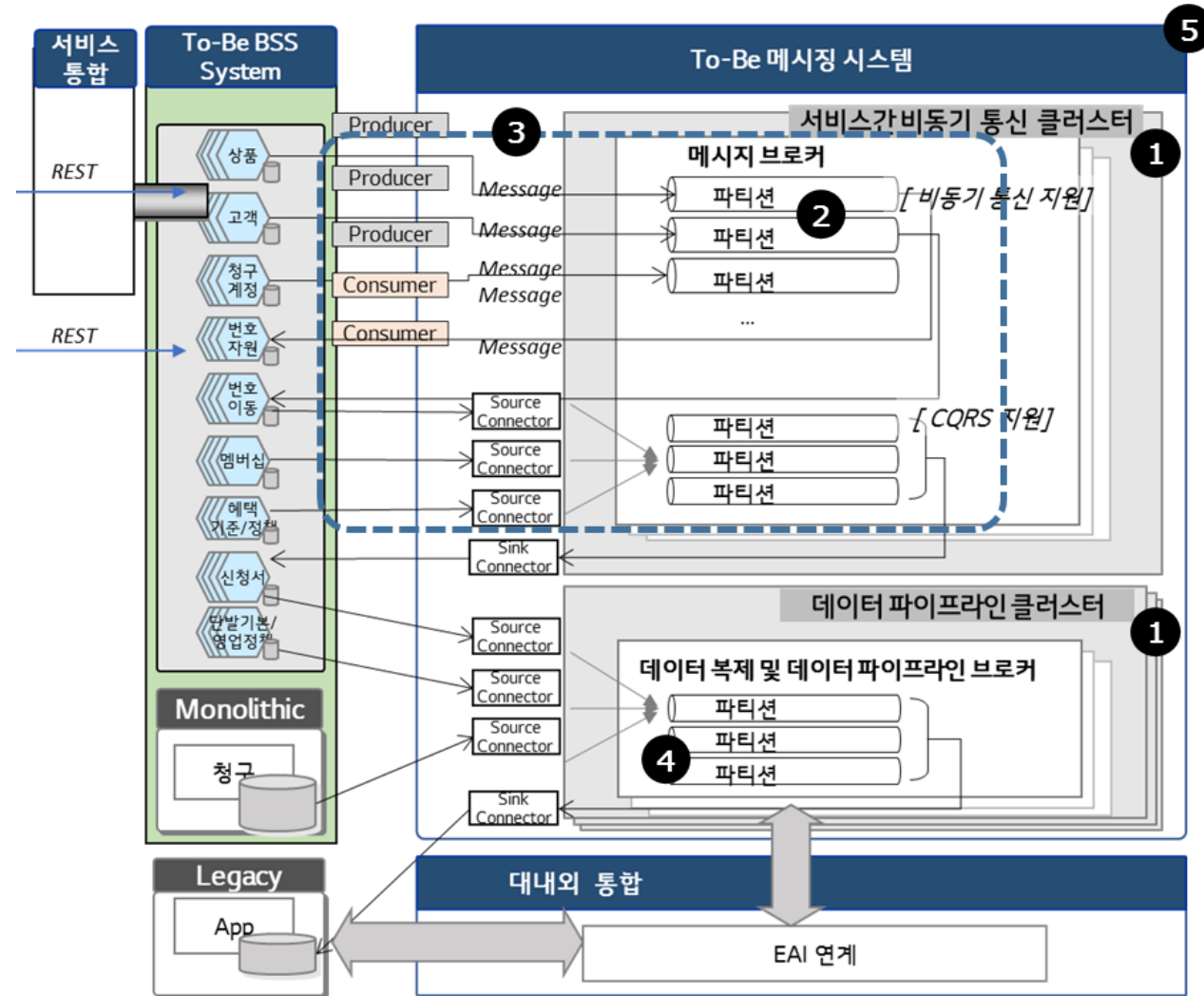
2 데이터 복제 및 파이프라인



- Kafka 중심의 데이터 파이프라인 구성
- DB2DB, 준실시간/배치 분석, 대용량 데이터 전송 업무에 적용

3.3 차세대 인프라 기술 – 비동기 메시징

□ 아키텍처 TO-BE 이미지



□ 메시징 구성 요약

1. 메시징 클러스터 구성

- 서비스간 비동기 통신 → 단일 통합 클러스터 사용
- 데이터 파이프라인 → 목적에 따른 클러스터 분할 사용

2. 토픽/파티션 구성

- 서비스 별 토픽 생성 (Request, Response, Event)
- 표준화된 공통 포맷 사용 (Header, Body, Json)

3. MSA 분산 전환에 따른 구조적 문제 대응

- 분산 데이터간 정합성 유지를 위한 메시징 지원
- 데이터 변경이력 유지, 복합 조회를 위한 메시징 지원

4. Kafka 기반 데이터 복제

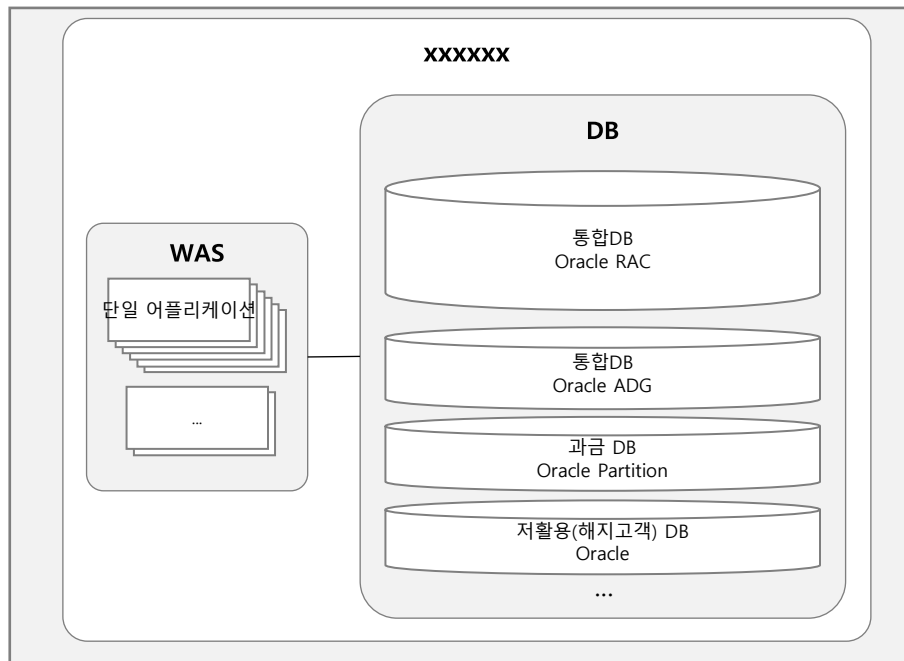
- Kafka 기반 데이터 파이프라인 통한 복제 방안 제시 (DB2DB, 준실시간/배치 분석, 대용량 데이터 전송)

5. 비기능 요건 수용

- 데이터 일관성 및 가용성 보증 방안 제시
- 서비스 확장성, 모니터링 방안 제시

3.3 차세대 인프라 기술 – 데이터 스토어

□ AS-IS (Monolithic) 현황



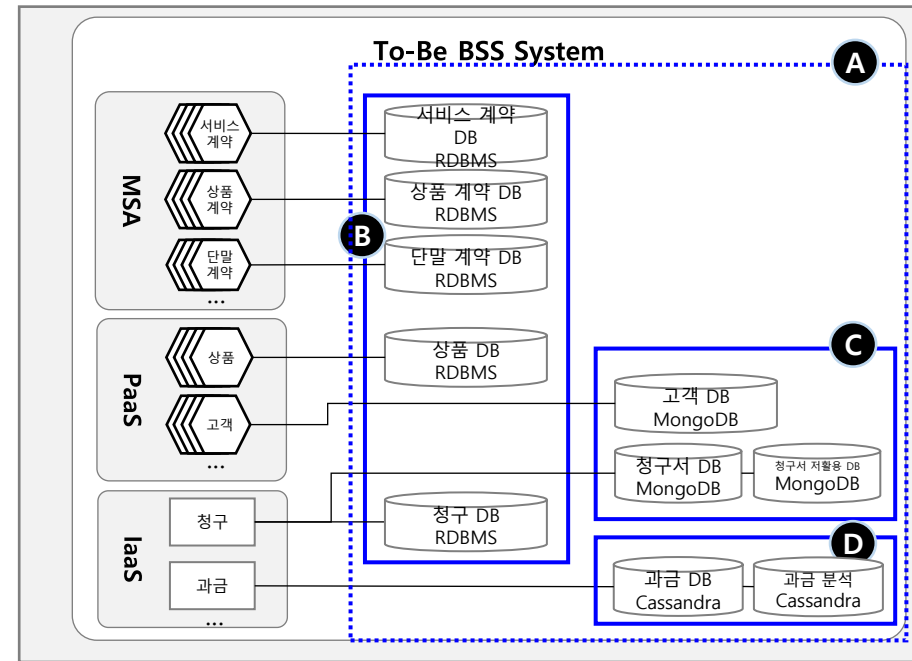
A 데이터베이스 분산

- 어플리케이션의 분산 영역 기준으로 DB도 분산 구성을 원칙으로 함
- 상호 연관성에 따라 복수 어플리케이션이 데이터를 상호 참조 할 수 있음

B RDBMS

- 업무특성 : 트랜잭션(데이터 정합성) 보장 필요, 패키지 솔루션 제약 등
- 적용기준 : 단순 조회업무는 오픈RDBMS(MariaDB), 동시변경 영역은 Oracle 사용

□ TO-BE 데이터베이스 아키텍처(분산 전환)



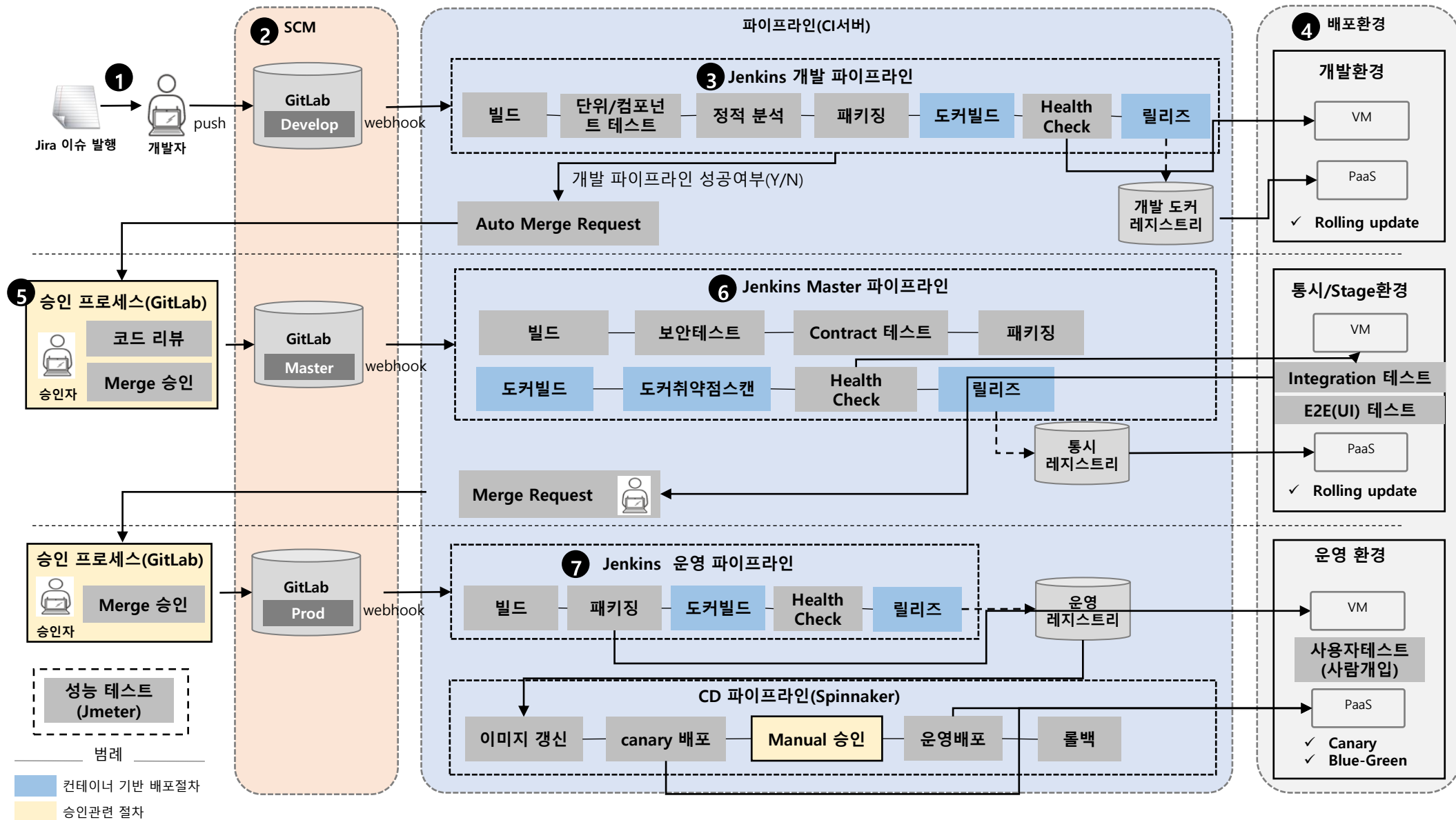
C MongoDB (문서 스토어)

- 업무특성 : 데이터 모델이 복잡하고 변경이 많은 영역
- 기대효과: 개발 생산성 향상과 개발 L/T감소로 시장 대응 속도 향상

D Cassandra (칼럼 스토어)

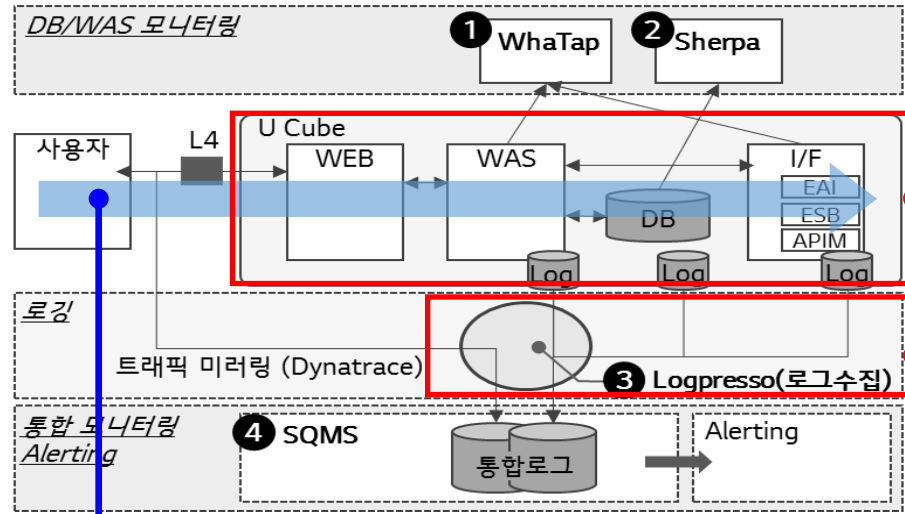
- 업무특성 : 대용량 데이터 처리량 증가에 대한 대비가 필요한 영역
- 기대효과: 향후 가입자 증가에 대비한 수평 성능 확장이 용이

3.3 차세대 인프라 기술 – CI/CD 자동화

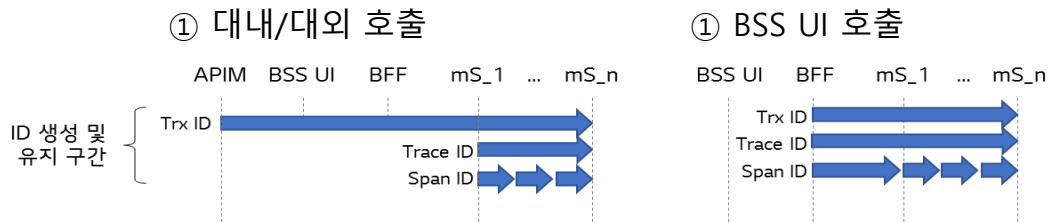


3.3 차세대 인프라 기술 – 통합모니터링

□ AS-IS (Monolithic)



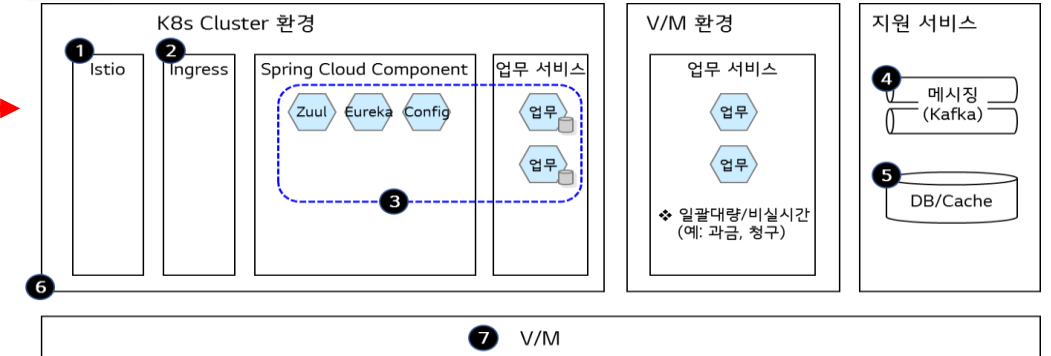
3 E2E 트랜잭션 추적



- 대내/대외, BSS UI 로 구분하여 E2E 트랜잭션 경로 분석
- { Trx ID, Trace ID, Span ID } 3개 ID 의 조합과 연계로 E2E 트랜잭션 추적
- Spring Cloud Sleuth, Zipkin 기반 추적 방안 설계

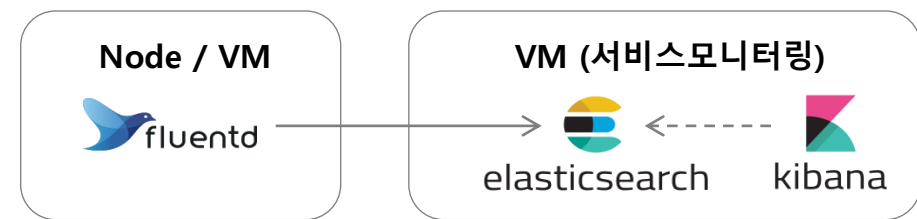
□ TO-BE (MSA 기반 분산 전환)

1 포괄적인 모니터링



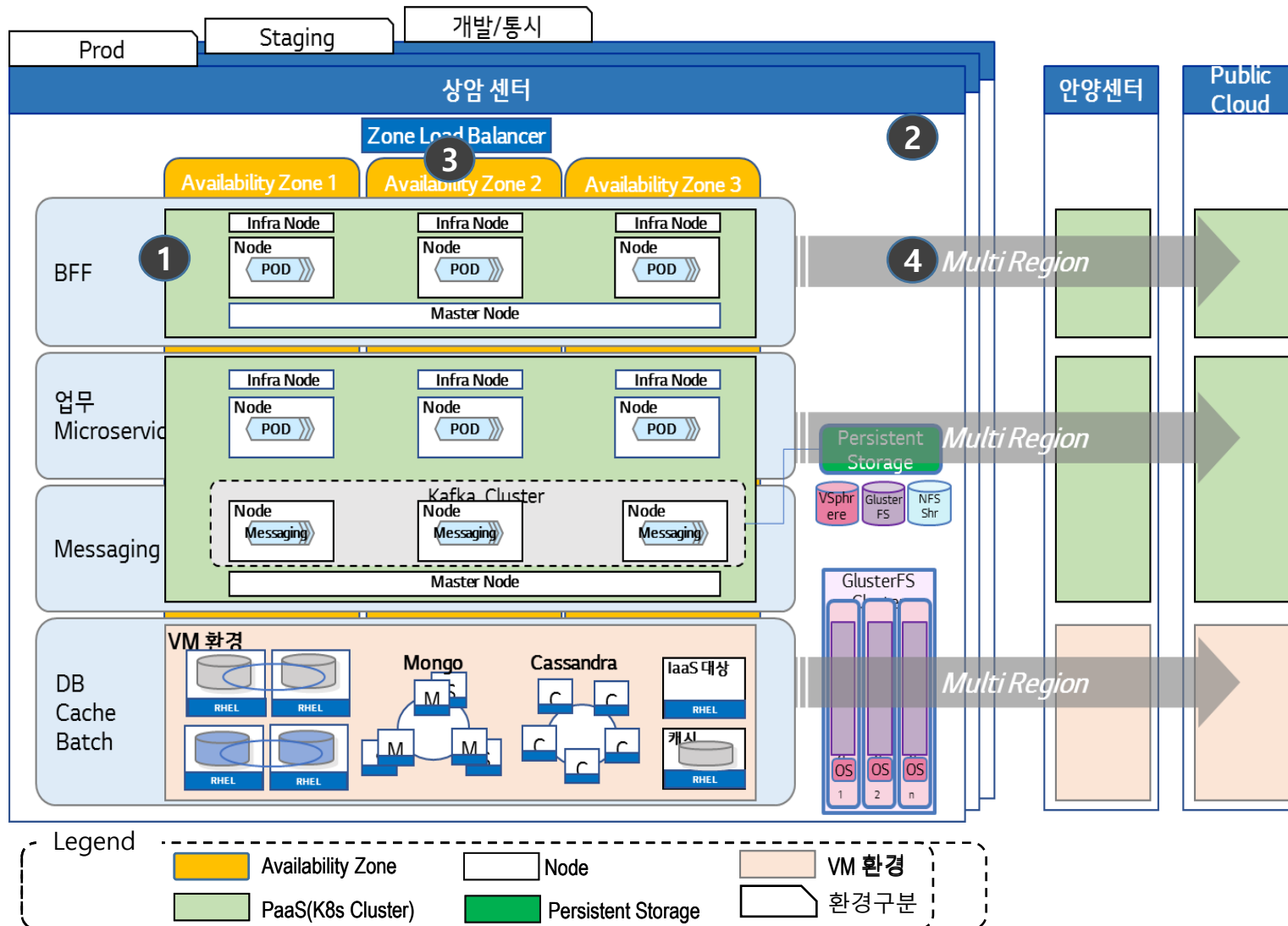
- E2E 관점에서 모니터링 대상이 되는 주요 기술 컴포넌트 식별
- Prometheus / Grafana 기반 모니터링 방안 설계

2 안정적인 통합 로깅



- K8s & VM 환경에서 대량 로그 데이터에 대한 안정적인 관리
- EFK 기반 통합 로깅 방안 설계

> To-Be BSS 서비스 운영 아키텍처



> 주요 설계 항목

1 센터 내 클러스터 구성방안

- 서비스메쉬 아키텍처와 메시징 아키텍처를 고려한 Kubernetes 클러스터 구성

2 Isolation 모델 정의

- 물리 기준 환경 분리 방안
 - 환경 분리 기준 정의하고 구성방안 수립
- 논리 기준 환경 분리 방안
 - 업무별, 솔루션 별 Namespace를 활용한 isolation 기준 정의
 - 네트워크 고려사항 정의

3 HA 구성 방안

- HA 모델 검토
- 차세대 BSS HA 구성 방안 및 장애 대응방안

4 다중 센터 지원 요구사항 정의

- 다중 센터를 위한 To-be 아키텍처 검토
- Hybrid Cloud 지원방안 검토