

코어뱅킹 분산전략

Dec. 14, 2017

bankware global

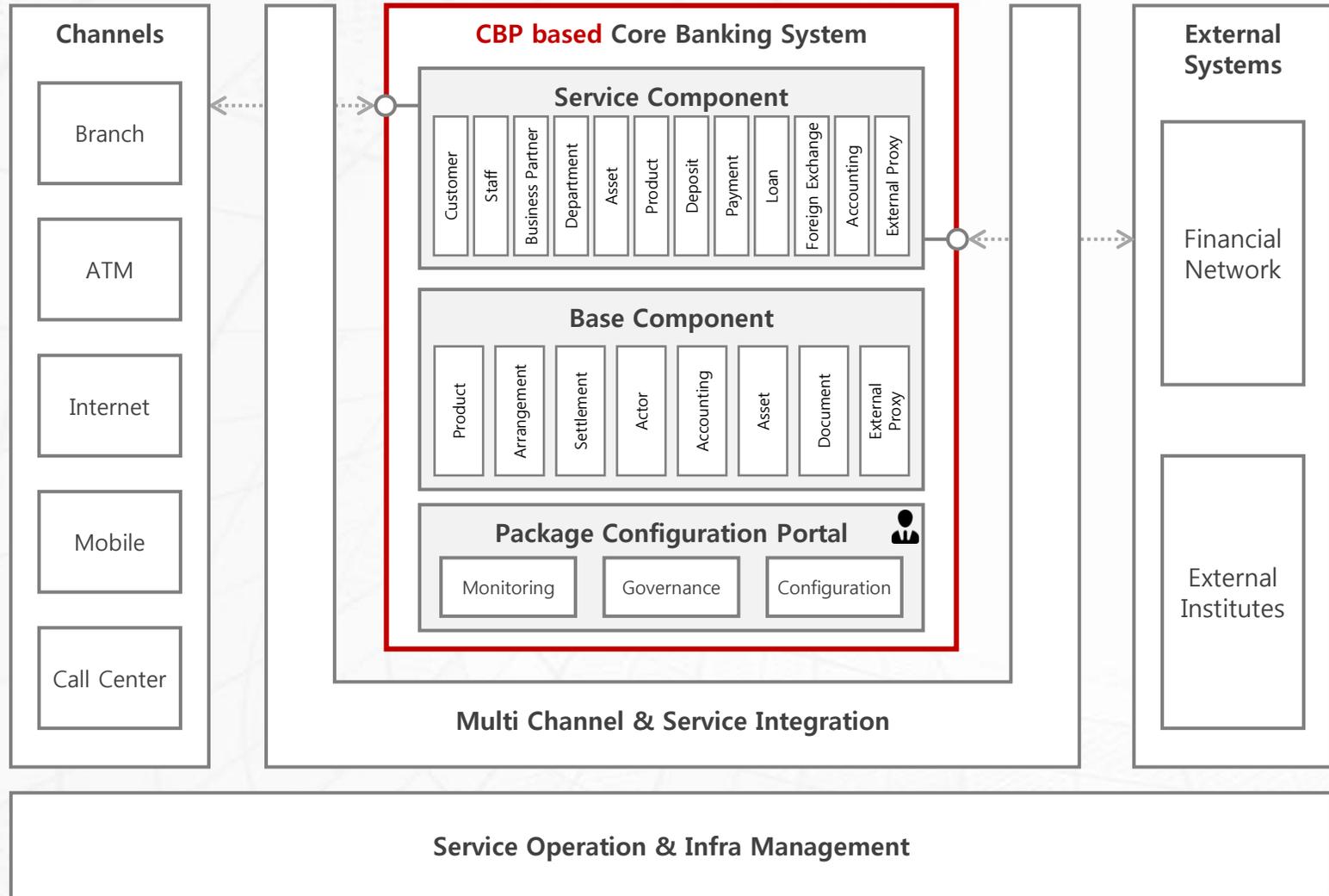
코어 बैं킹 모델

유연성

확장성

표준화

모듈화

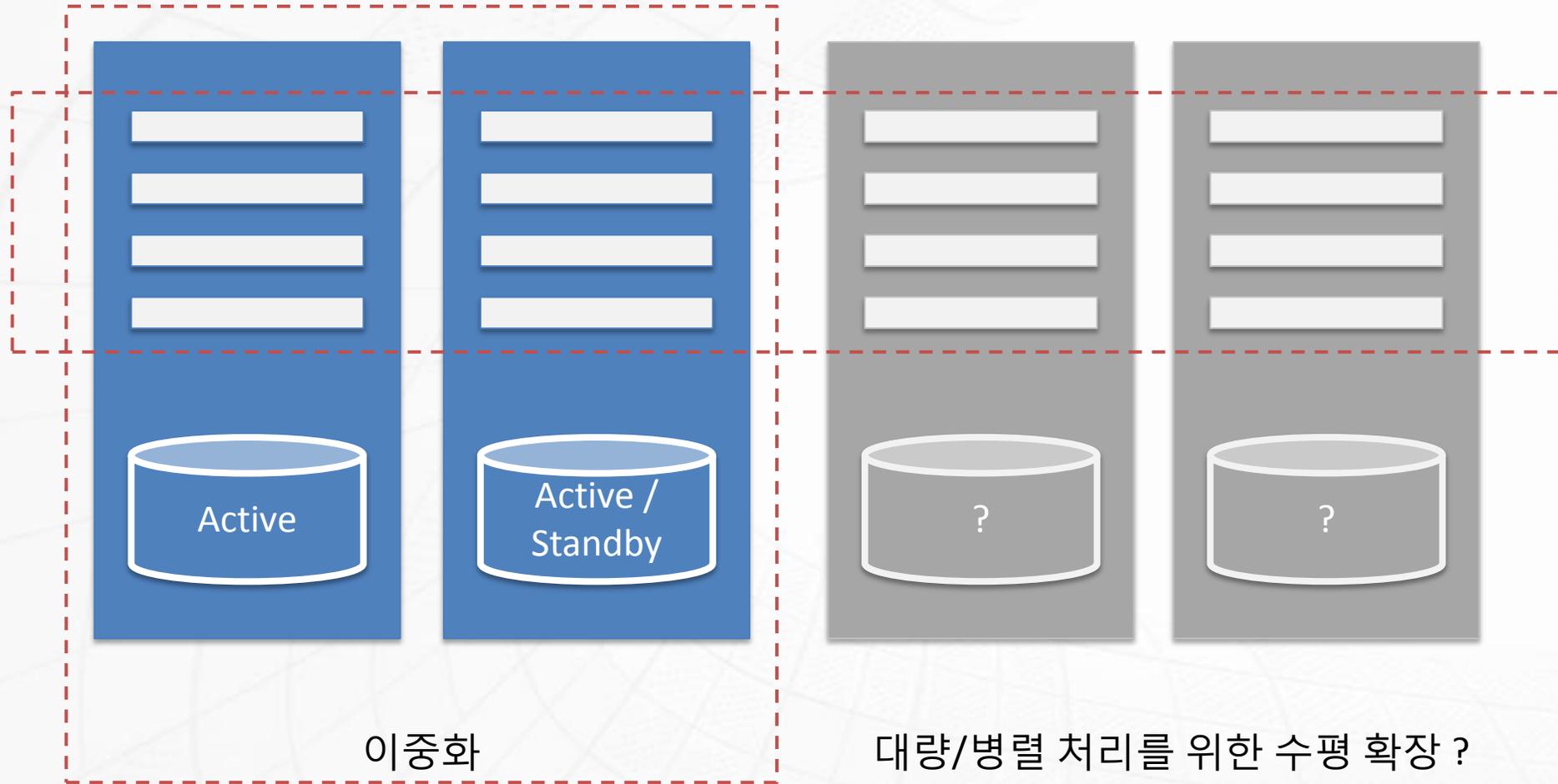


Assemblable

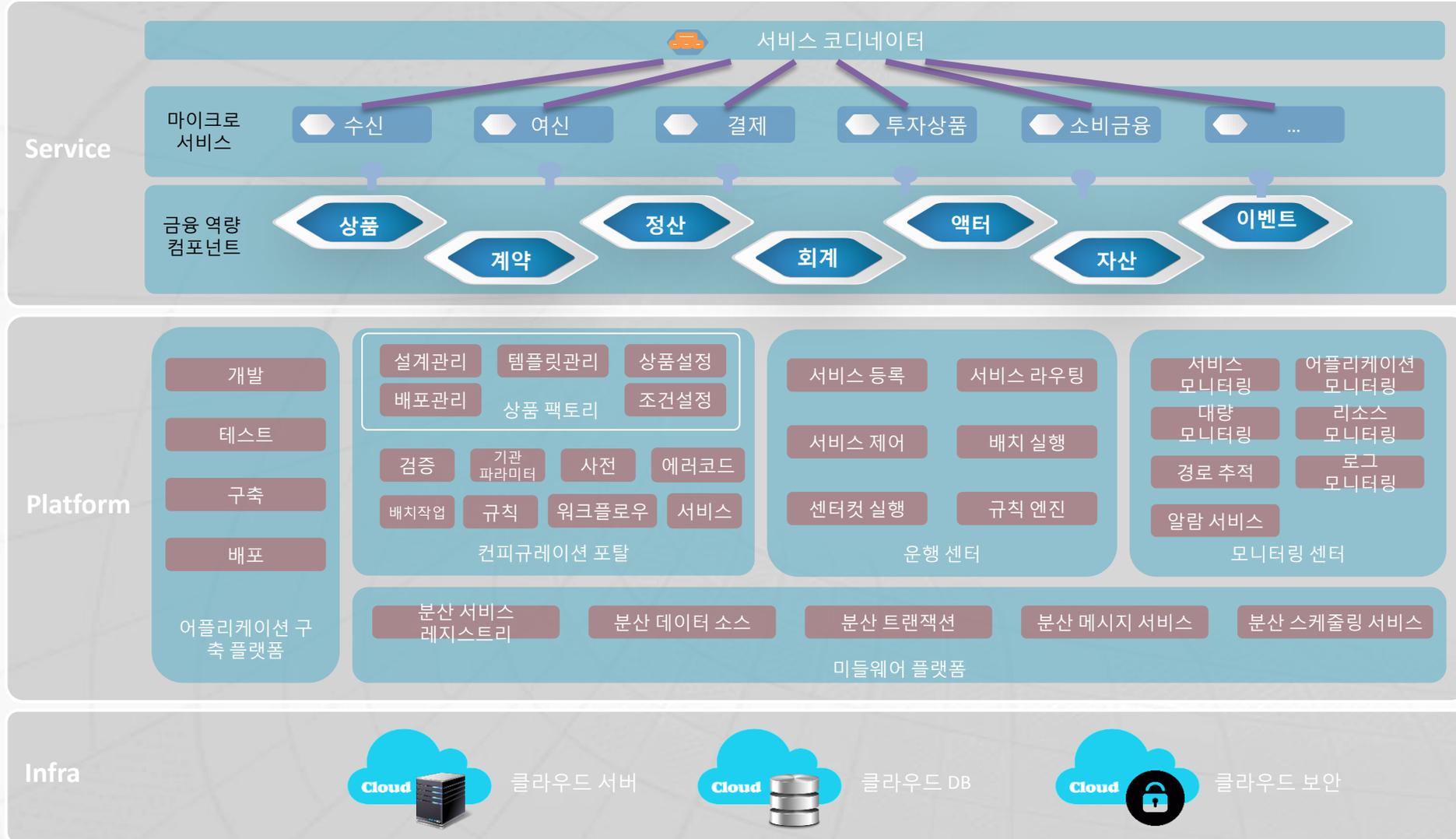
Pluggable

Configurable

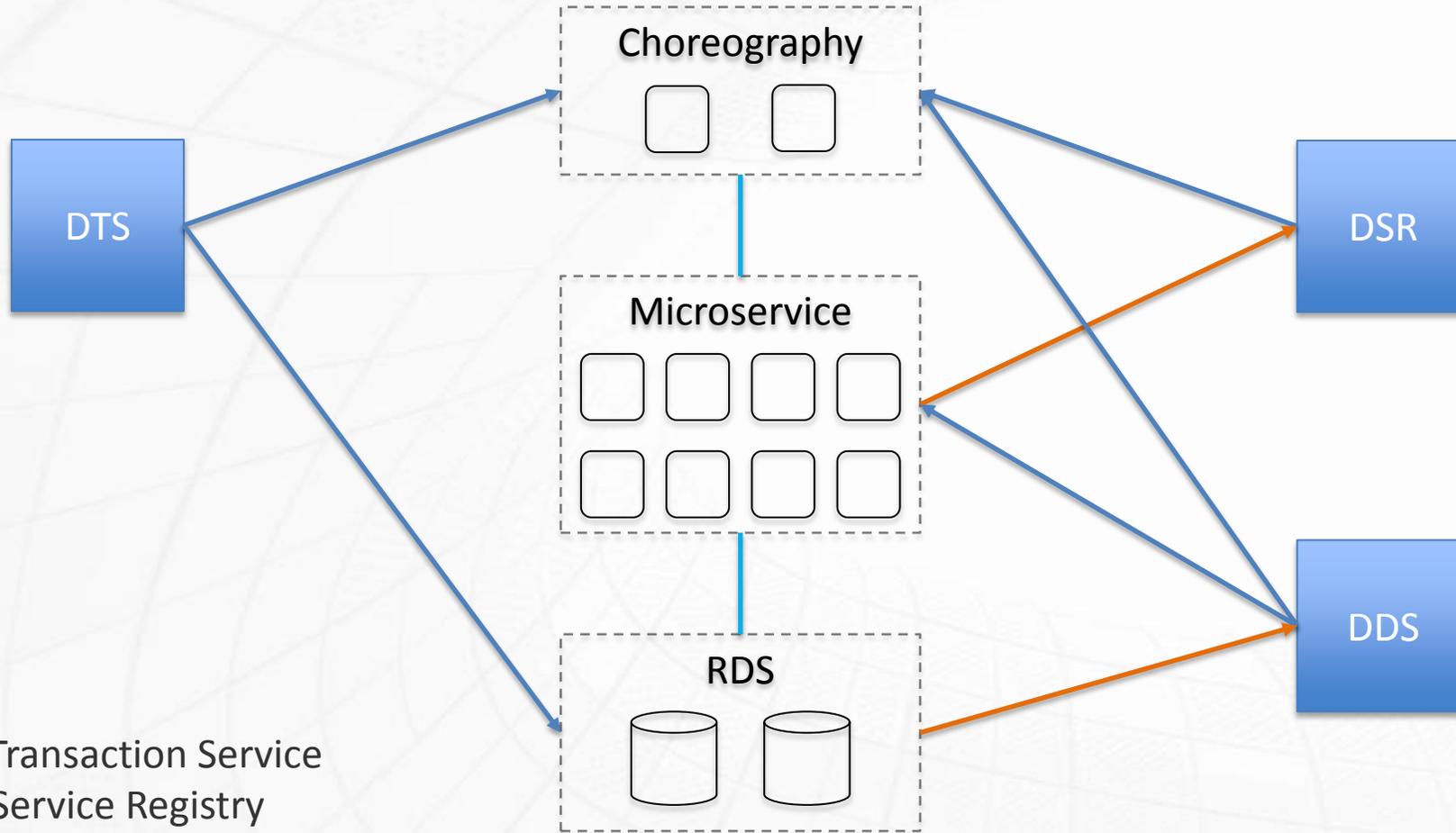
코어 बैं킹 시스템 구성



클라우드를 이용한 분산 코어의 기술적 고려사항



Ant Financial Cloud를 이용한 시스템 구성

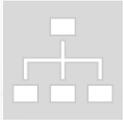


DTS : Distributed Transaction Service
DSR : Distributed Service Registry
DDS : Distributed DataSource Service

기술적 고려사항 - 마이크로 서비스 구조



업무와 함께 성장하는 민첩성



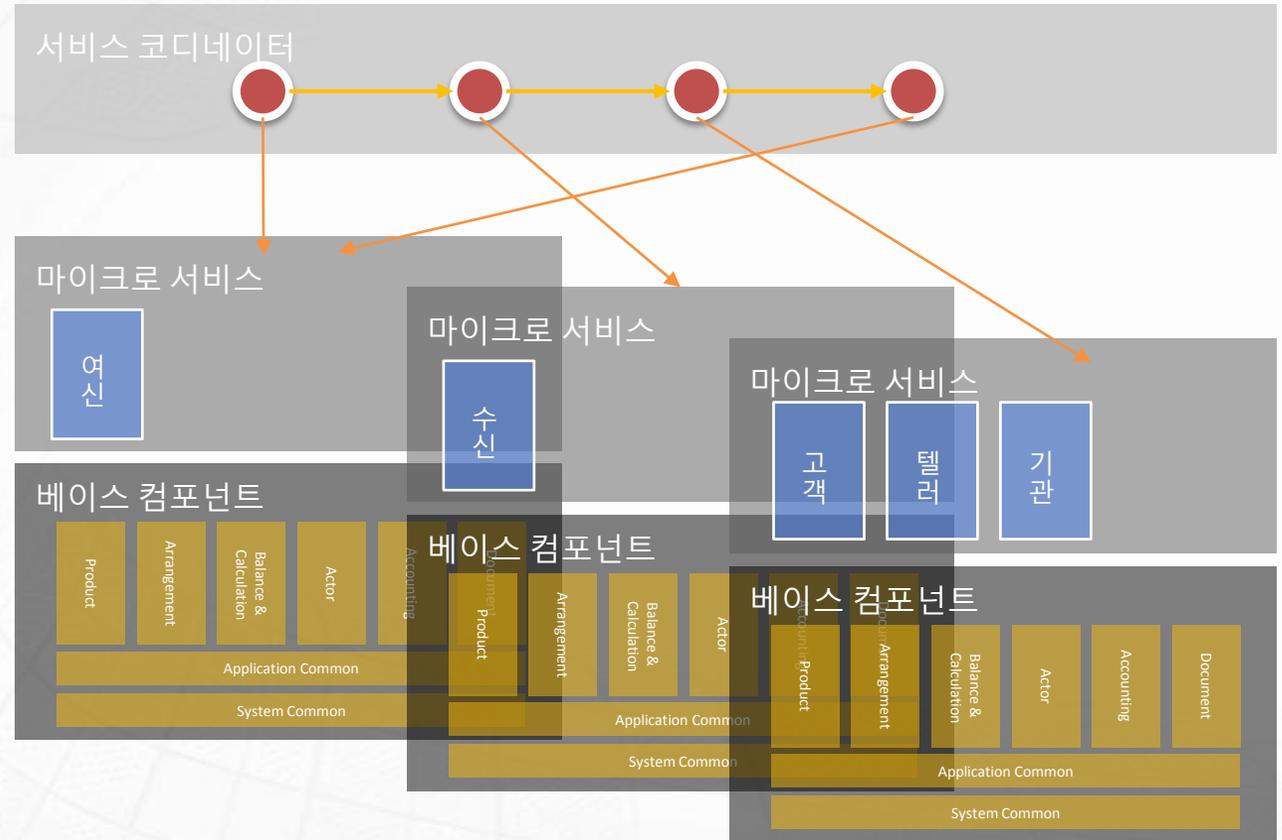
계층형 아키텍처

- ✓ 서비스 코디네이터
- ✓ 마이크로 서비스
- ✓ 베이스 컴포넌트



어플리케이션 단위

- ✓ 보편 타당한 업무구성
- ✓ 업무주관부서의 시각
- ✓ 자유로운 조합 배치



기술적 고려사항 - 트랜잭션의 관리



2 Phase Commit:

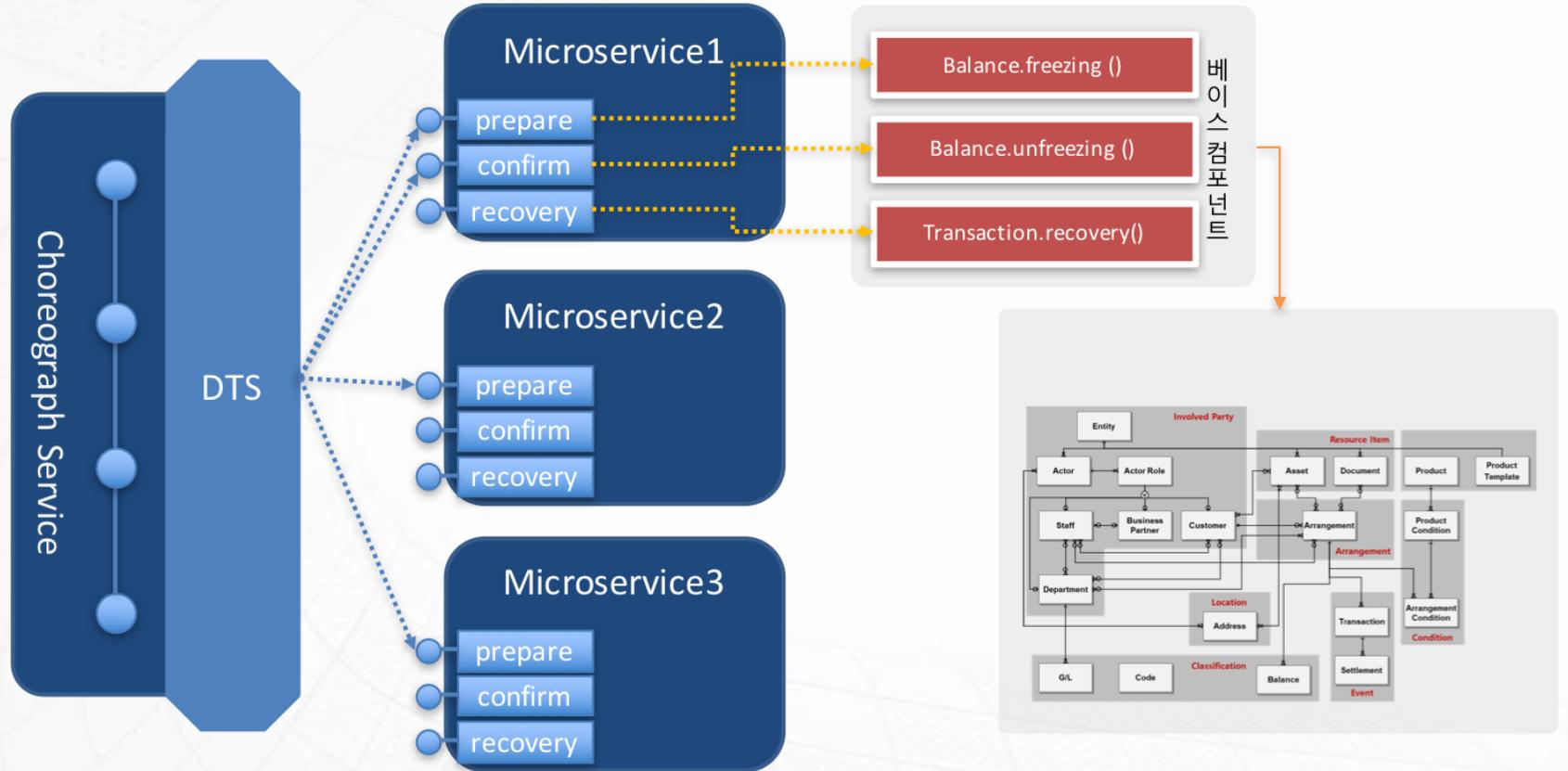
분산된 마이크로 서비스들의 개별적 commit 데이터의 처리



분산 트랜잭션 서비스:

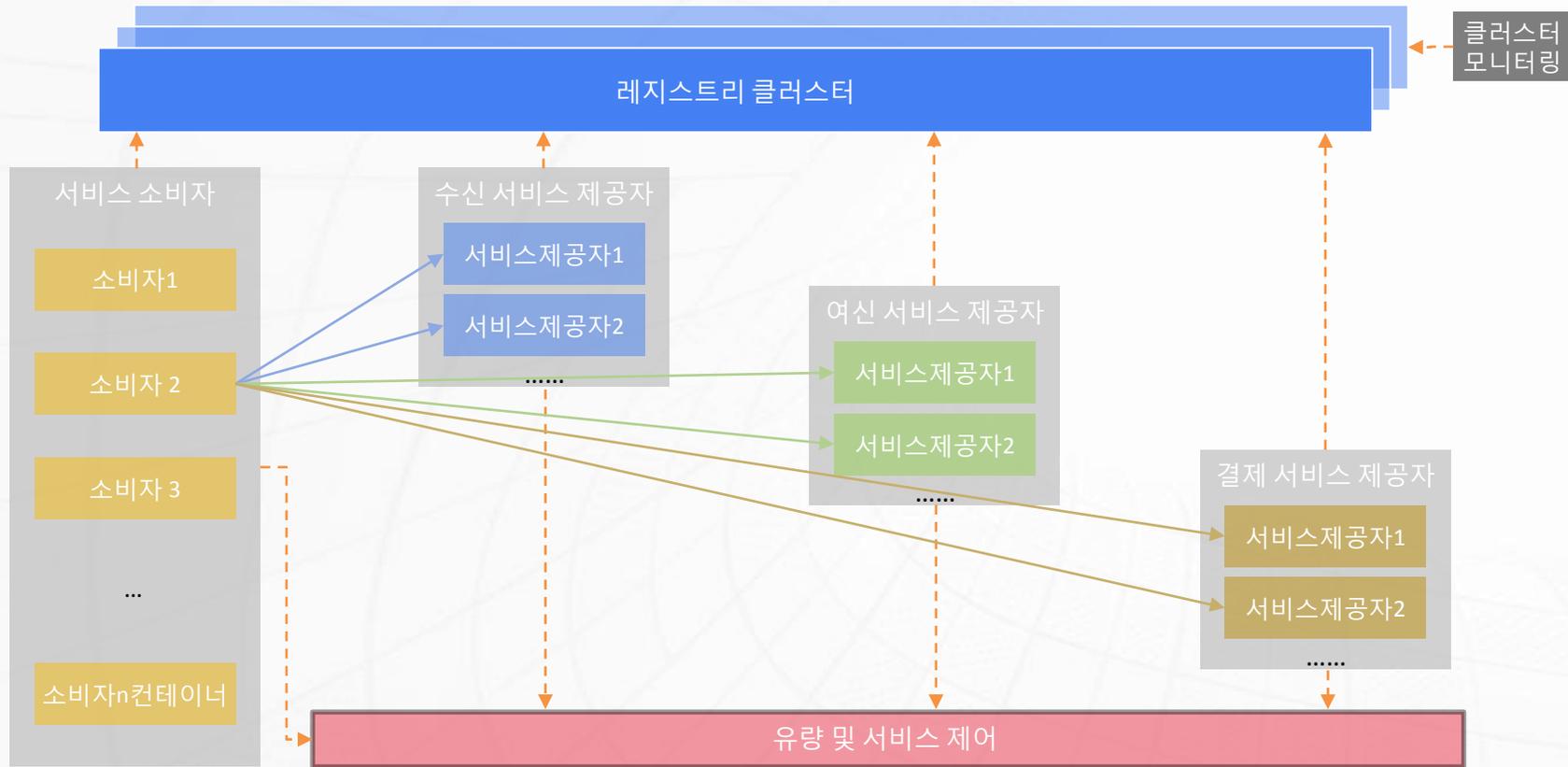
XA 같은 데이터베이스 의존이 아닌 프로그램으로 commit 과 rollback에 해당하는 confirm과 recovery 구현.

데이터 모델에 대응하는 컴포넌트의 freezing / unfreezing / recovery 구현으로 필요한 데이터 처리 일원화



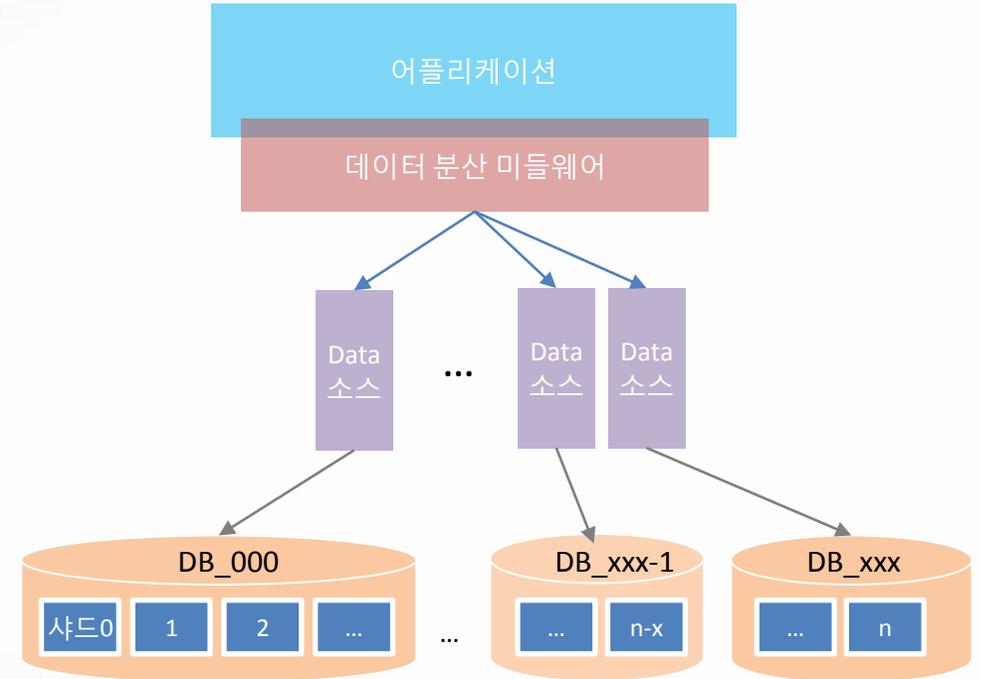
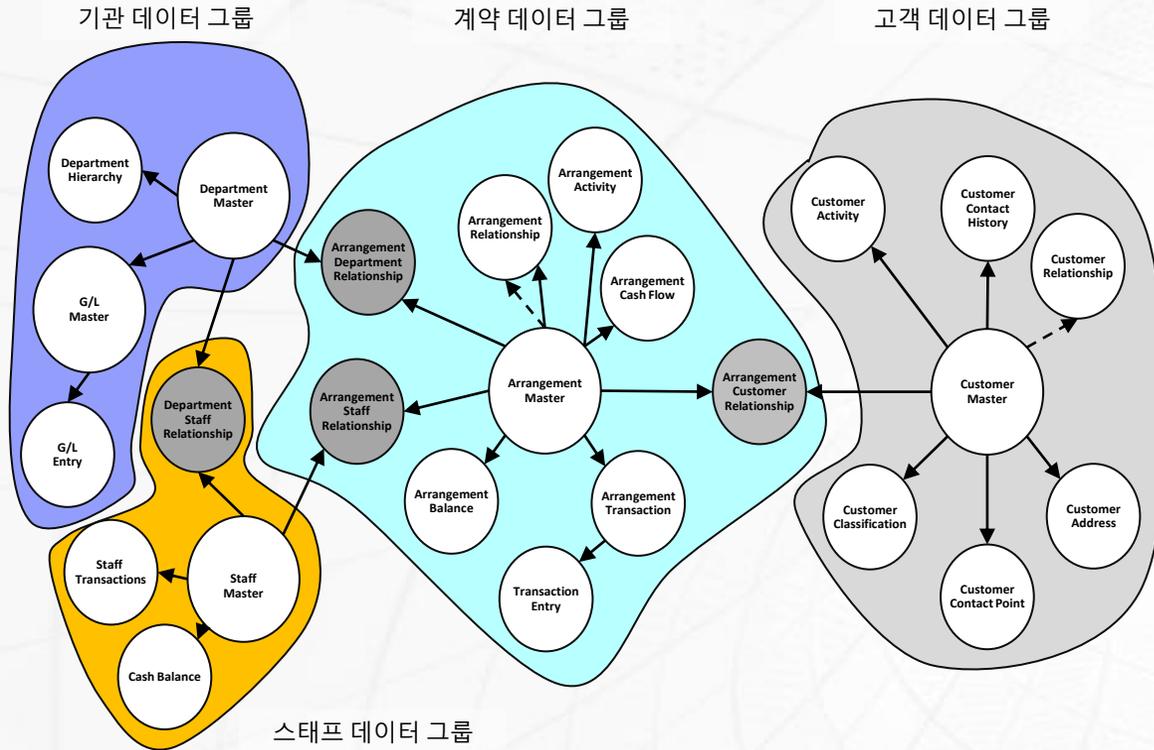
기술적 고려사항 - 동적 서비스 관리

엔트파이낸셜의 마이크로 서비스 프레임워크로 서비스의 전방위 관리



- ✓ 동적 서비스 등록 및 발견
- ✓ 전 호출 경로 추적
- ✓ 소프트 로드 전략 정의
- ✓ 유량 및 서비스 제어
- ✓ 블루그린 배포

기술적 고려사항 - 데이터의 분산



- ✓ 데이터 그룹 별 배치 전략 수립 (단일 / 샤딩 테이블)
- ✓ 샤딩키 정의 (ex- Business Object Primary Key)

- ✓ 테이블 샤딩 룰 정의
- ✓ 분산 데이터 소스 정의

성능 테스트 결과

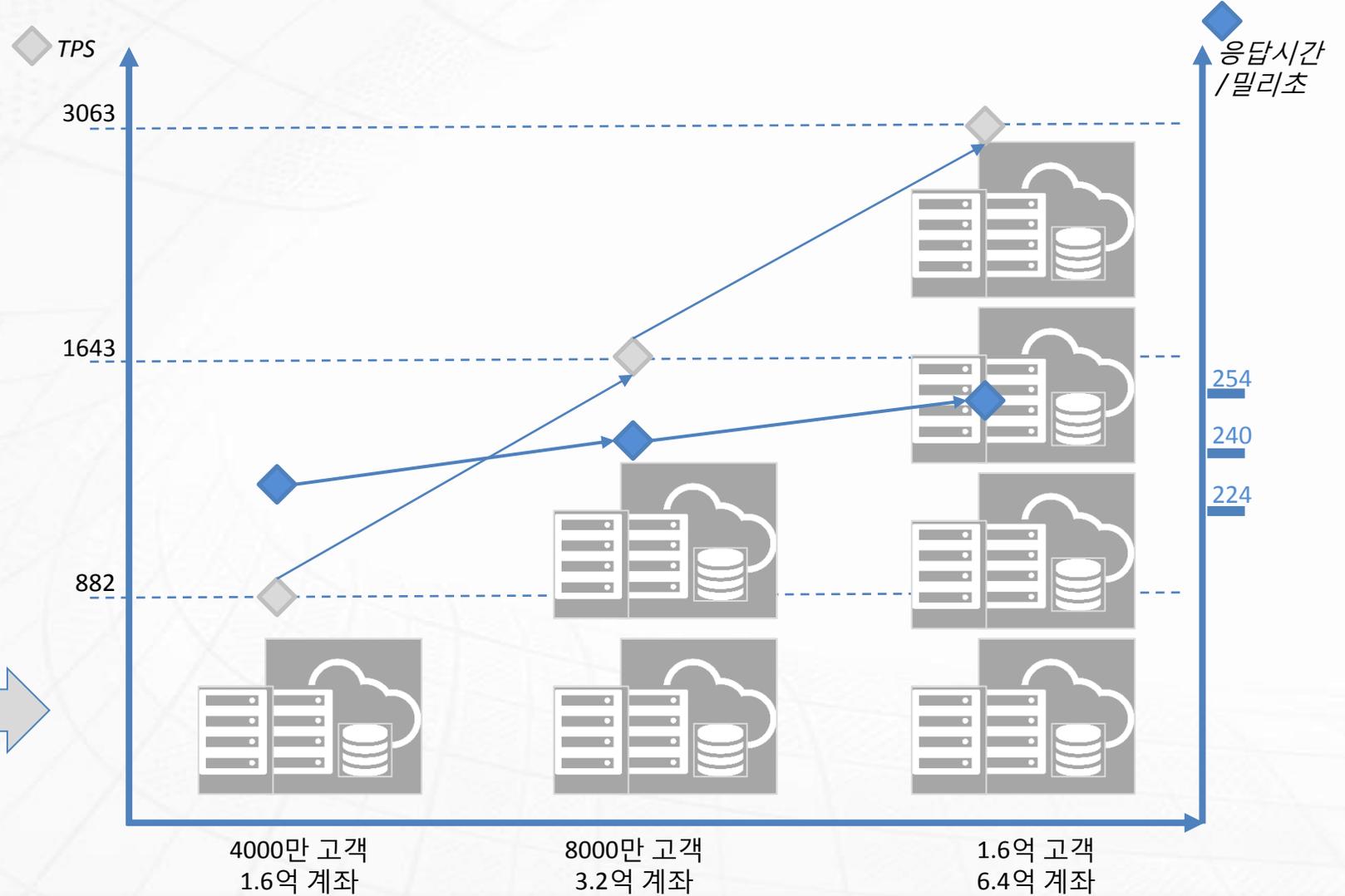
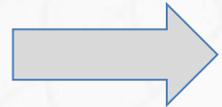


집중 아키텍처(기존):
Scale Up 방식으로 고객 규모 증가에 대응. 한계 비용 상승과 한계 효율 하락의 문제 발생.



분산 아키텍처:
Scale Out으로 고객 규모 대응. 안정적인 응답 환경에서 TPS 선형적 증가 가능.

- ✓ 출금
- ✓ 입금
- ✓ 거래내역조회
- ✓ 로그인
- ✓ 스마트예금입금
- ✓ 스마트예금출금
- ✓ 신용대출신규
- ✓ 상환스케줄조회
- ✓ 연결카드리스트조회
- ✓ 상품별잔액조회
- ✓ 계좌기본정보조회
- ✓ 이체



차세대 코어가 분산 기술을 만났을 때

