

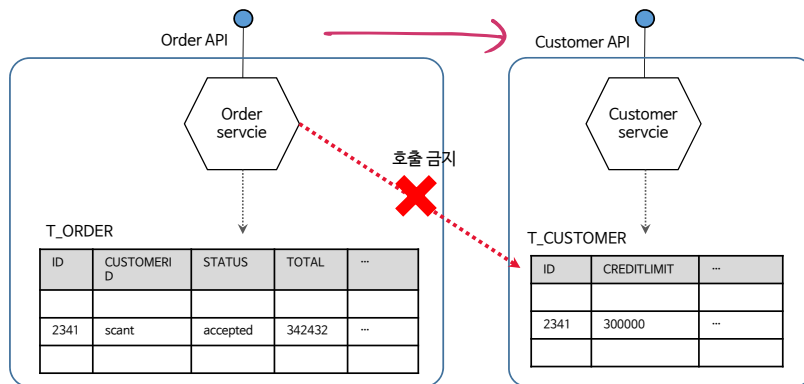
Microservice 트랜잭션 처리 - SAGA 패턴

SK(주) C&C 한정현

서비스별 데이터베이스

- Microservice Application 데이터베이스 Architecture
- 각 마이크로 서비스의 영구 데이터를 다른 서비스에 대해 비공개로 유지하고 API를 통해서만 액세스

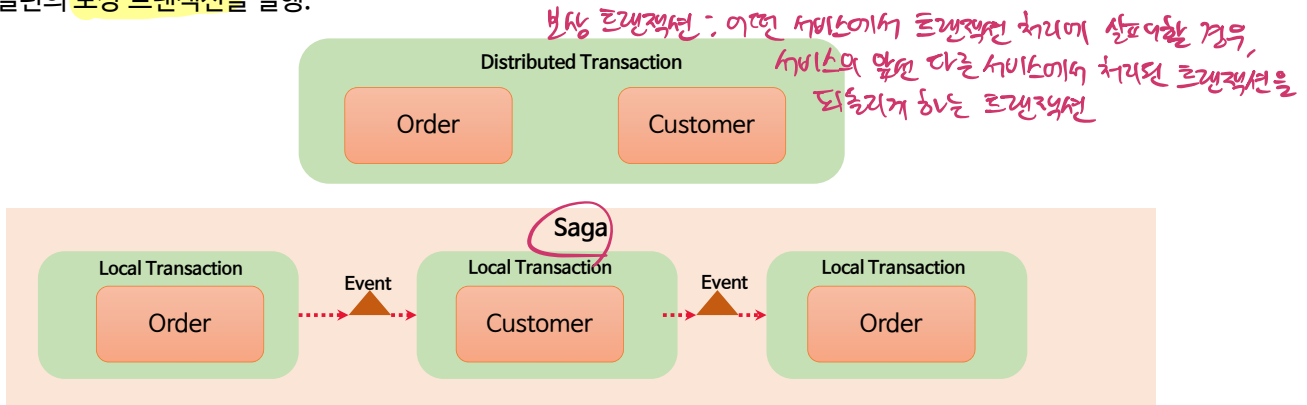
바로 질의 불가. API를 통해서 접근 가능



보안 트랜잭션 문제 발생 가능

saga

- 여러 서비스간에 데이터 일관성을 유지하는 방법
- 사가 (saga): 로컬 트랜잭션을 순차적으로 처리하는 패턴. 각 로컬 트랜잭션은 자신의 데이터베이스를 업데이트하고 사가내에 있는 다음 로컬 트랜잭션을 트리거하는 메시지 또는 이벤트를 게시.
- Biz 규칙을 위반하여 로컬 트랜잭션이 실패한 경우, saga는 이전 로컬 트랜잭션이 작성한 변경 사항을 실행 취소하는 일련의 보상 트랜잭션을 실행.

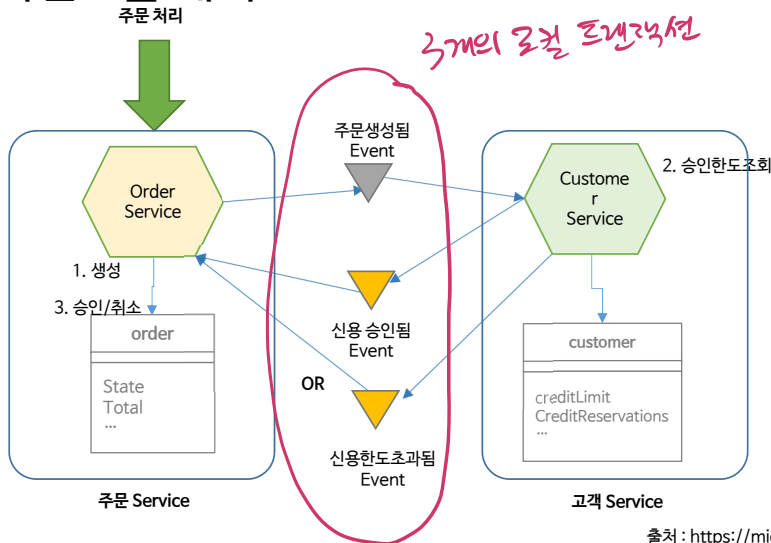


주문처리시 고객의 신용한도 정보에 따라 최종 주문을 승인하는 업무

↳ 2가지 Saga 패턴 적용 가능

Saga - Choreography

- 구성 : 각 로컬 트랜잭션은 다른 서비스의 로컬 트랜잭션을 트리거하는 Domain 이벤트를 게시.



Saga - Orchestrator

이벤트를 통해서가 아니라, Orchestration이 직접 처리

