

도메인 주도 개발 (DDD)

• [DDD란](#)

DDD란

DDD(Domain-Driven Development)는 2003년 에릭 에반스가 Domain-Driven Design이라는 책을 처음 출간하면서 소개한 개발 방법론으로, "훌륭한 소프트웨어를 개발하고 싶다면 서비스 도메인에 귀를 기울여라"라는 슬로건으로부터 시작되었고, 현재는 서비스 개발에 가장 큰 주류를 이루고 있는 개발 방법입니다.

오늘날 가장 보편화된 시스템 아키텍처인 MSA(Micro Service Architecture)를 구현하는 필수 개념들은 DDD로부터 왔는데, DDD에서 좋은 서비스를 개발하기 위한 핵심 기본 요소인 Loose Coupling(느슨한 결합)과 High Cohesion(높은 응집)은 MSA를 설계할 때 꼭 기억해야 할 설계 원칙입니다.

DDD의 주요 설계 원칙: Loose Coupling(느슨한 결합)과 High Cohesion(높은 응집)

그럼 무엇을 Loose Coupling하고 무엇을 High Cohesion 해야 할까요? 바로 도메인입니다. **도메인들 간에는 Loose Coupling하고 도메인 내에서는 High Cohesion 해야 합니다.** 도메인은 소프트웨어로 해결하고자 하는 문제의 영역, 즉 개발하고자 하는 전체 서비스를 잘라낸 단위를 가리키는데요, 도메인을 잘못 나누면 DDD나 MSA 입장에서는 많은 혼란이 가중됩니다. 왜냐하면, Loose Coupling 해야 하는 연동 인터페이스를 High Cohesion 하게 되어 시스템 복잡도를 높이거나, High Cohesion 해야 할 서비스들 간을 Loose Coupling 해서 예상하지 못한 시스템 문제를 야기할 수 있기 때문입니다.

따라서, 도메인을 잘게 나누는 것, 즉 Loose Coupling 시키는 것만이 능사가 아니라, 어떤 서비스들을 하나의 도메인으로 잘 묶어서 High Cohesion 하게 할지 설계하는 것까지가 DDD나 MSA가 추구하는 지향점이 되어야 합니다. 즉, **도메인을 Loose Coupling과 High Cohesion 관점에서 잘 나누는 것이 DDD와 MSA에서 가장 중요하다** 할 수 있습니다. 비즈니스 문제를 잘 투영한 서비스 도메인을 잘 나누는 것에서부터 시작하며, 각 도메인 서비스들이 Loose Coupling과 High Cohesion 각각을 지원할 수 있는 기술적 또는 아키텍처적 설계 원칙을 준수하는 것이 좋은 서비스 시스템을 개발하는 기본 원칙입니다.

Loose Coupling과 High Cohesion 설계 원칙이 잘 적용된 서비스 도메인을 도식화하면 다음과 같습니다.

