

클라우드 임팩트 2020 버추얼 컨퍼런스 (11.10 ~ 11.11)

금융사 사례로 이해하는
마이크로 서비스 접근방안 (Mono to Micro)

Nov. 2020
임용성 상무

IBM GARAGE PRACTICE LEADER

CLOUD INTEGRATION EXPERT SERVICE LEADER

*Microservices are a software development technique
- a variant of the service oriented architecture (SOA) architectural style that
structures an application as a collection of loosely coupled services. ...
Microservice-based architectures enable continuous delivery and deployment*

- 사례 이해 및 마이크로 서비스 필요성
- 사례로 이해하는 마이크로 서비스 접근 방안
- 활용 가능한 오픈링 및 도구, 그리고 권고사항

새로운 서비스 출시에 대한 고민

보험 서비스를 새롭게 출시를 계획하고 있으며
새로운 서비스 빠르게 개발을 해야하고,
신기술에 대한 적용이 필요하며,



경쟁력 있고 차별화된 서비스에 대한 개발
및 운영이 가능해야 함

주요 의사결정 사항

1. 무엇을 해야 하는가?

- 모놀리틱 어플리케이션 설계/개발
- 마이크로 서비스 설계/개발
- 온프레미스 환경 검토 및 도입
- 클라우드 플랫폼 선정 및 도입

2. 어떻게 만들어야 하는가?

- 외부 전문 사업자에 개발 외주화
- 외부 전문가와 Co-Work을 통한
- 역량 내재화로 주도적 설계/개발
- 현업 사용자의 적극적 참여 확보

기대 효과

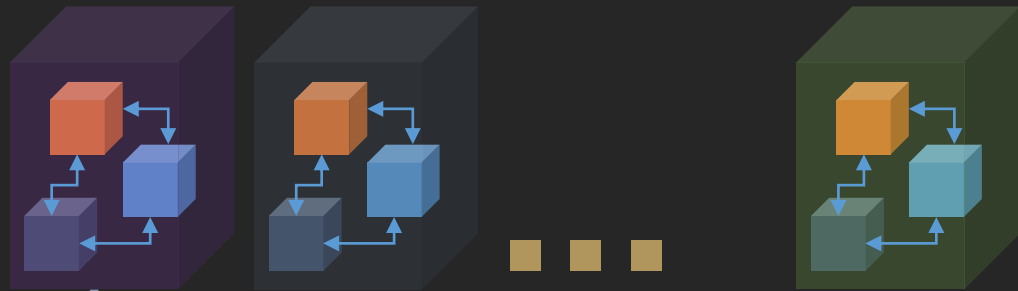
마이크로 서비스 도입을
통해서 ...

- 서비스의 빠른 출시가
가능하고,
- 개별 서비스의 경쟁력
제고가 쉬움

역량 내재화를 통한 현업과
함께 주도적 설계 및
개발을 통해서 ...

- 외부 종속성을 탈피할
수 있으며,
- 자체적으로 아이디어
발굴 및 적용을
빠르게 할 수 있음

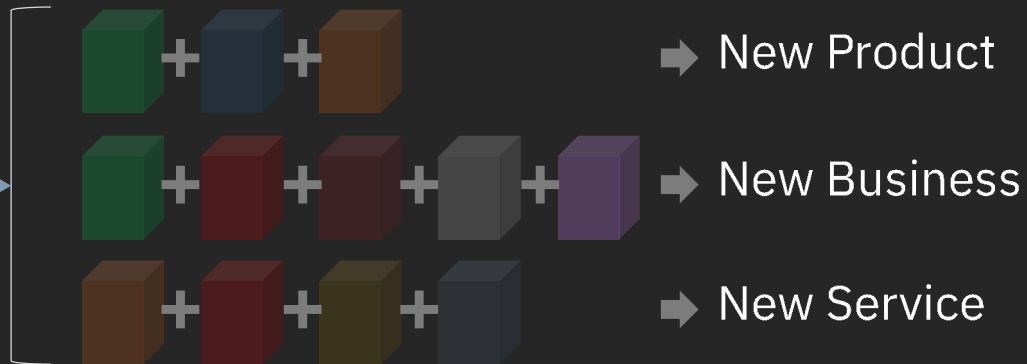
비즈니스 역량 기반의 마이크로 서비스로 전환 필요



Microservice (마이크로 서비스)는

- 독립적이며 의미 있는 최소 단위의 비즈니스 역량
- **Single unit of business capability or responsibility**

Easy to use & quick



독립된 비즈니스 역량 (혹은 마이크로 서비스)은

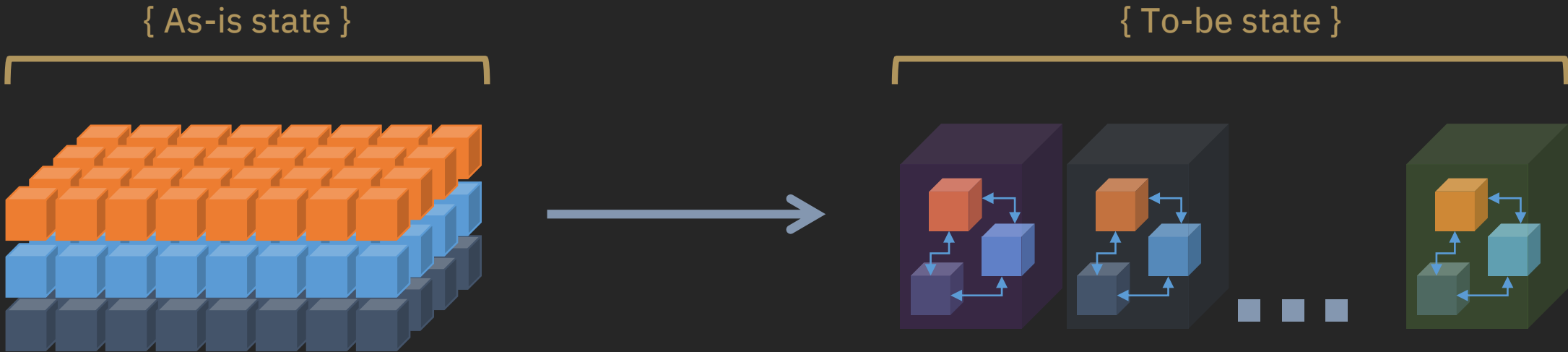
- 새로운 상품 혹은 새로운 비즈니스 모델, 새로운 서비스를 출시하는데 활용될 수 있는 최소의 단위이며,
- 독립적 개발 및 운영에 따라 비즈니스 증속성 없이 쉽고 빠르게 활용이 가능함

클라우드 도입과 함께 통합 아키텍처에서 분산 아키텍처로 변화



새로운 디지털 기술의 출현으로 급격하게 변화하는 비즈니스 환경에 빠르게 대응하기 위해서는

- 독립적으로 특화된 비즈니스 역량 (혹은 서비스)을 제공할 수 있어야 하며
- 새로운 기술 (인공지능, 블록체인, 클라우드, 데이터, 모바일 등)을 빠르게 활용할 수 있어야 함



- 통합 아키텍처를 기반으로 표준화, 통합화를 통해 효율적인 운영에 관심
- 각각의 비즈니스 역량 (혹은 서비스)의 경쟁력은 다소 약화될 수 있음

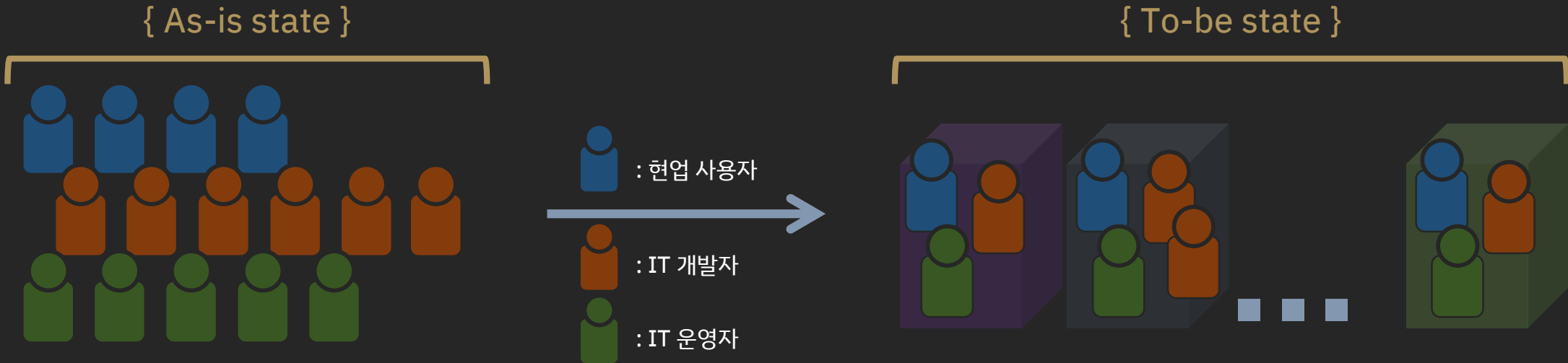
- 분산 아키텍처를 기반으로 비즈니스 역량 (혹은 서비스) 단위의 신속한 개발 및 운영에 관심
- 각각의 비즈니스 역량 (혹은 서비스)의 경쟁력을 높일 수 있으며 개발/운영에 특화된 기술을 적용할 수 있음

분산 아키텍처를 개발 및 운영하기 위한 조직 구조의 변화 필요



독립적인 단위의 분산된 마이크로 서비스 아키텍처를 효율적으로 개발 및 운영하기 위해서는

- 마이크로 서비스 단위의 개발/운영 중심의 프로세스 및 조직을 구성하고,
- 비즈니스 도메인 전문가부터 개발 및 운영에 관련된 모든 이해당사자를 하나의 팀 (Squad)으로 구성함



- 표준화된 통합 환경을 효율적으로 개발 및 운영하기 위해서는 분업화된 조직 및 프로세스 구성이 필요함
- 개발 및 운영 생산성을 높일 수 있으며 비용 효율화 하는데 기여할 수 있음

- 독립된 서비스 단위를 빠르게 개발 및 운영하기 위해서는 서비스 단위로 조직 및 프로세스의 구성이 필요함
- 서비스에 특화된 개발 및 운영 환경을 적용할 수 있기 때문에 빠르고 경쟁력 있는 서비스에 최적화할 수 있음

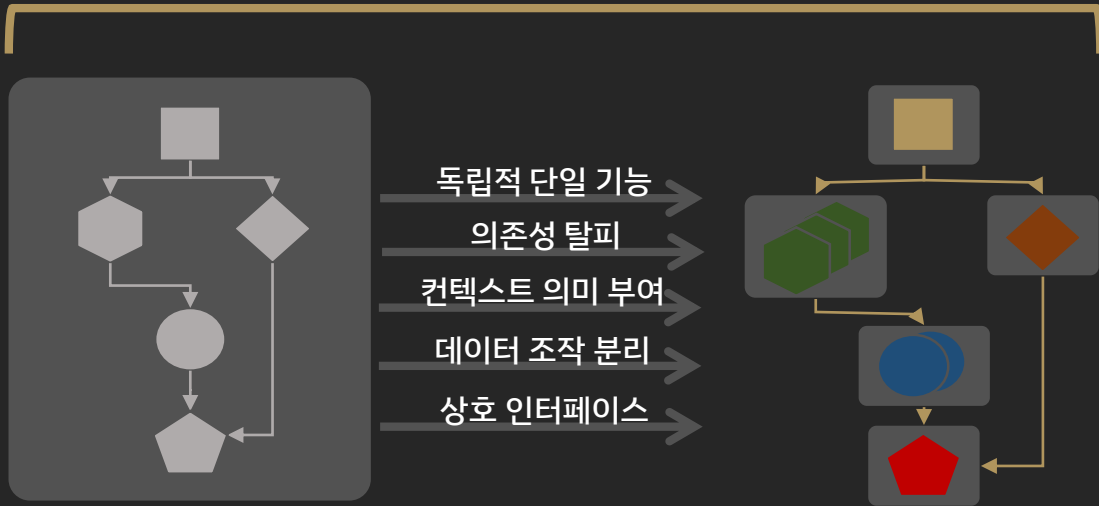
고객 사례에 대한 배경 이해



생명 보험 비즈니스를 수행하는 고객사에서는

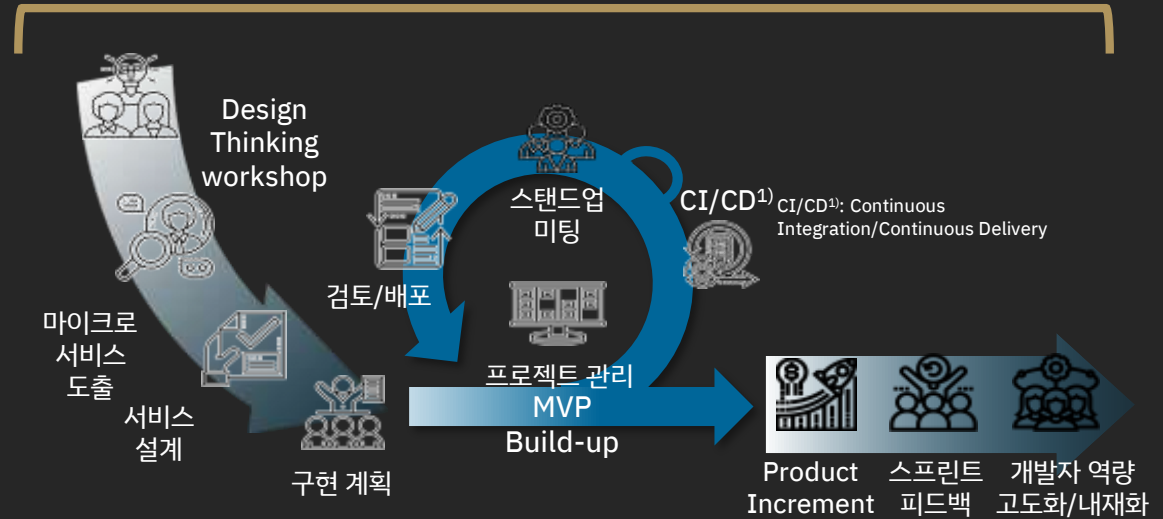
- 새로운 온라인 보험 서비스를 클라우드 기반 마이크로 서비스로 빠르게 출시하고,
- 현업과 IT의 긴밀한 협업과 새로운 기술에 대한 필수 역량의 내재화를 통해서 주도적으로 개발 및 운영을 하려고 함

마이크로 서비스로 경쟁력 확보



- 온라인 보험 서비스를 위해 필요한 비즈니스 역량을 마이크로 서비스 중심으로 식별, 설계 및 개발/운영을 통해서
- 빠르게 서비스를 출시하고 경쟁력을 높이고자 함

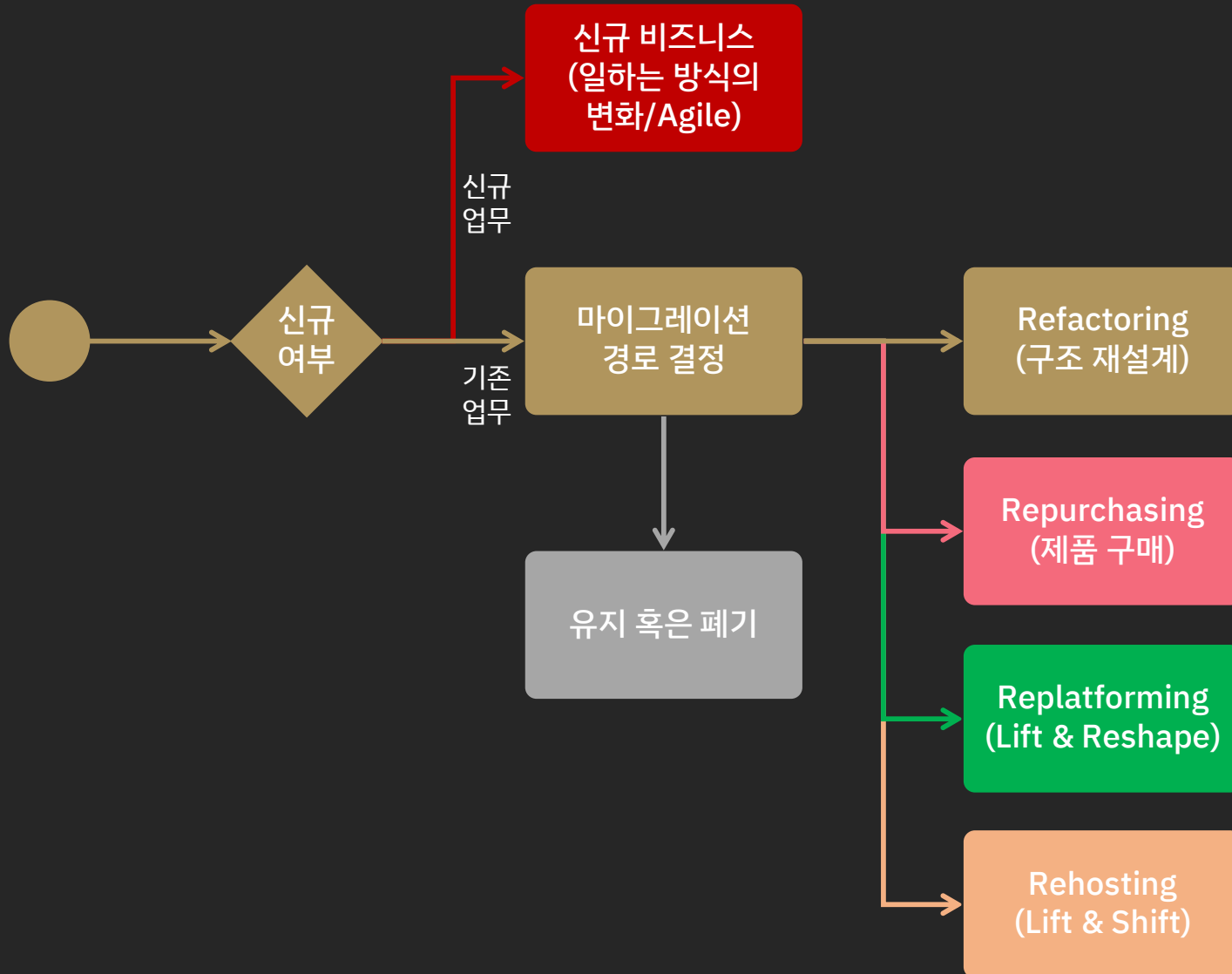
역량 내재화 기반 주도적 개발 및 운영



- 서비스 기획부터 출시, 그리고 운영까지의 전체 과정의 필수 역량 내재화를 통해서 종속성 탈피 및 신속한 개발 필요함

- 사례 이해 및 마이크로 서비스 필요성
- 사례로 이해하는 마이크로 서비스 접근 방안
- 활용 가능한 오픈링 및 도구, 그리고 권고사항

클라우드 및 마이크로 서비스아키텍처 로 전환을 위한 로드맵



- 아이디어/아키텍처 구성 및 개발
- 역량 내재화 기반 주도적 수행 필요

- 어플리케이션 현대화를 위한 서비스
- 기존 어플리케이션 분석 자동화 필요

- 어플리케이션 현대화를 위한 서비스
- 기존 어플리케이션 분석 자동화 필요

- 어플리케이션 마이그레이션 서비스
- 기존 어플리케이션 분석 자동화 필요

GUIDE 1 독립적이고 유연한 기능 중심의 마이크로 서비스를 정의하고 독립적으로 개발 및 배포할 수 있는 서비스 식별 및 설계에 대한 구체적인 방안 및 방법론

METHOD: **마이크로 서비스를 어떻게 식별 및 설계 할 것인가?**

GUIDE 2 마이크로 서비스 단위로 데이터를 분리하기 위한 데이터 모델 구성 방안 및 서비스의 특성에 따른 데이터 솔루션 (NOSQL)의 적용 방안 및 가이드

DATA: **데이터 모델을 어떻게 정의하고 적용 솔루션은?**

GUIDE 3 마이크로 서비스의 개발 및 운영 증가에 따라서 개발 및 운영의 효율성을 보장하고 지속하기 위해서 필요한 프로세스 및 조직 등에 대한 방안

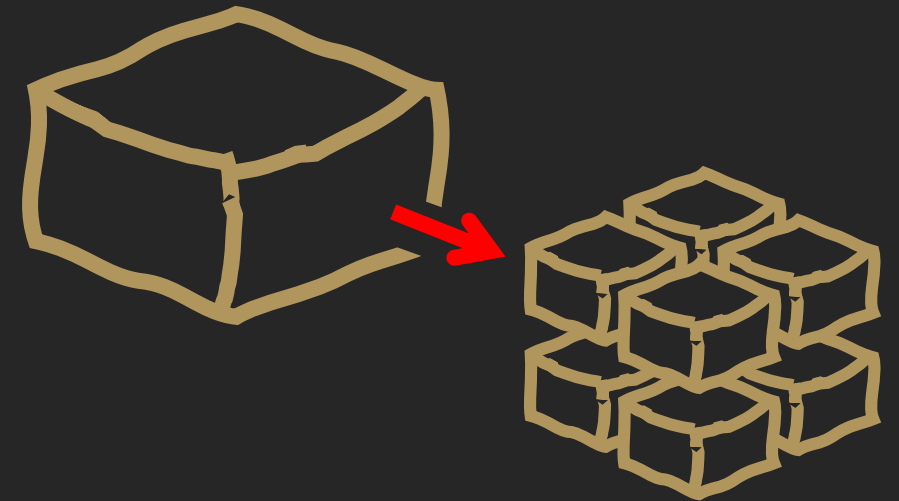
DEVOPS: **어플리케이션을 어떻게 개발 및 운영할 것인가?**

GUIDE 4 마이크로 서비스 아키텍처를 구현 및 배포/운영하기 위한 클라우드 플랫폼에 대한 아키텍처 및 고려사항

PLATFORM: **어플리케이션을 어디에 배포 및 운영할 것인가?**

GUIDE 5 마이크로 서비스 아키텍처 및 클라우드 플랫폼에 대한 필수 역량을 어떻게 준비하고 지속적으로 적용하기 위해서 필요한 역량 내재화 방안

CAPABILITY: **지속적인 개발 및 운영을 위한 필요 역량 및 주체는?**



GUIDE 1

독립적이고 유연한 기능 중심의 마이크로 서비스를 정의하고 독립적으로 개발 및 배포할 수 있는 서비스 식별 및 설계에 대한 구체적인 방안 및 방법론

METHOD: **마이크로 서비스를 어떻게 식별 및 설계 할 것인가?**

GUIDE 2

마이크로 서비스 단위로 데이터를 분리하기 위한 데이터 모델 구성 방안 및 서비스의 특성에 따른 데이터 솔루션 (NOSQL)의 적용 방안 및 가이드

DATA: **데이터 모델을 어떻게 정의하고 적용 솔루션은?**

GUIDE 3

마이크로 서비스의 개발 및 운영 증가에 따라서 개발 및 운영의 효율성을 보장하고 지속하기 위해서 필요한 프로세스 및 조직 등에 대한 방안

DEVOPS: **어플리케이션을 어떻게 개발 및 운영할 것인가?**

GUIDE 4

마이크로 서비스 아키텍처를 구현 및 배포/운영하기 위한 클라우드 플랫폼에 대한 아키텍처 및 고려사항

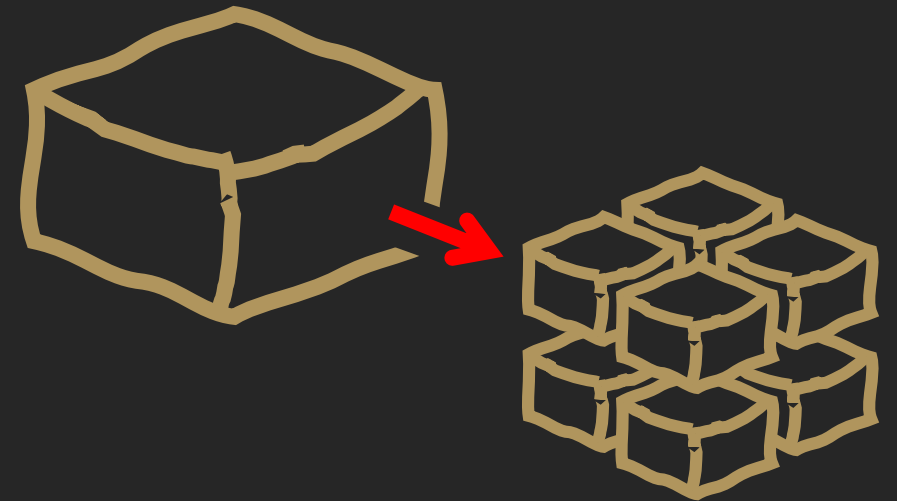
PLATFORM: **어플리케이션을 어디에 배포 및 운영할 것인가?**

GUIDE 5

마이크로 서비스 아키텍처 및 클라우드 플랫폼에 대한 필수 역량을 어떻게 준비하고 지속적으로 적용하기 위해서 필요한 역량 내재화 방안

CAPABILITY: **지속적인 개발 및 운영을 위한 필요 역량 및 주체는?**

🔗 **도메인 주도 설계 (Domain Driven Design)**
🔗 **Bounded Context & Ubiquitous Language**



마이크로 서비스 설계를 위한 전략적 접근 방안



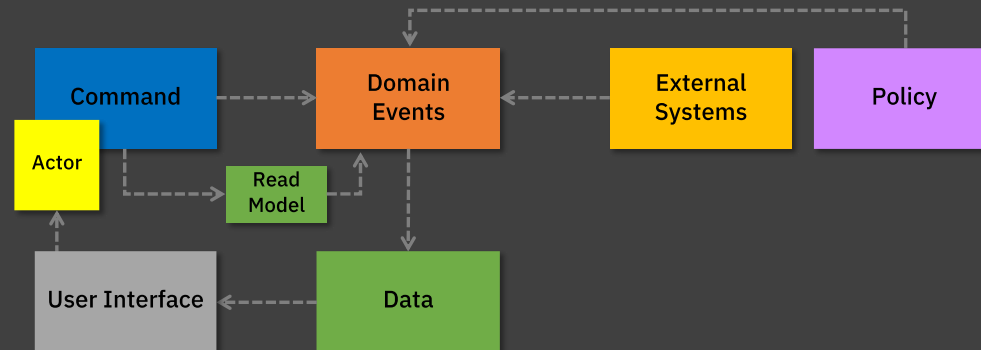
업무 도메인 중심의 분해

- 분석 대상이 되는 도메인을 정의함 (최상위 도메인 확정)
- 도메인 전문가와 IT 전문가의 협업을 통해 분해 및 정의함
- 업무 도메인에 대한 공동의 이해 및 공통의 용어 사용이 가능하게 됨

도메인 분석 결과를 참조하여 설계 일정을 수립함

도메인 기반 컨텍스트를 기준으로 독립적인 경계의 정의

- 도메인 이벤트를 중심으로 커맨드, 사용자, 정책 및 규칙, 외부 시스템, 관련 데이터에 대한 정의와 협의를 통해서 바운디드 컨텍스트를 정의함



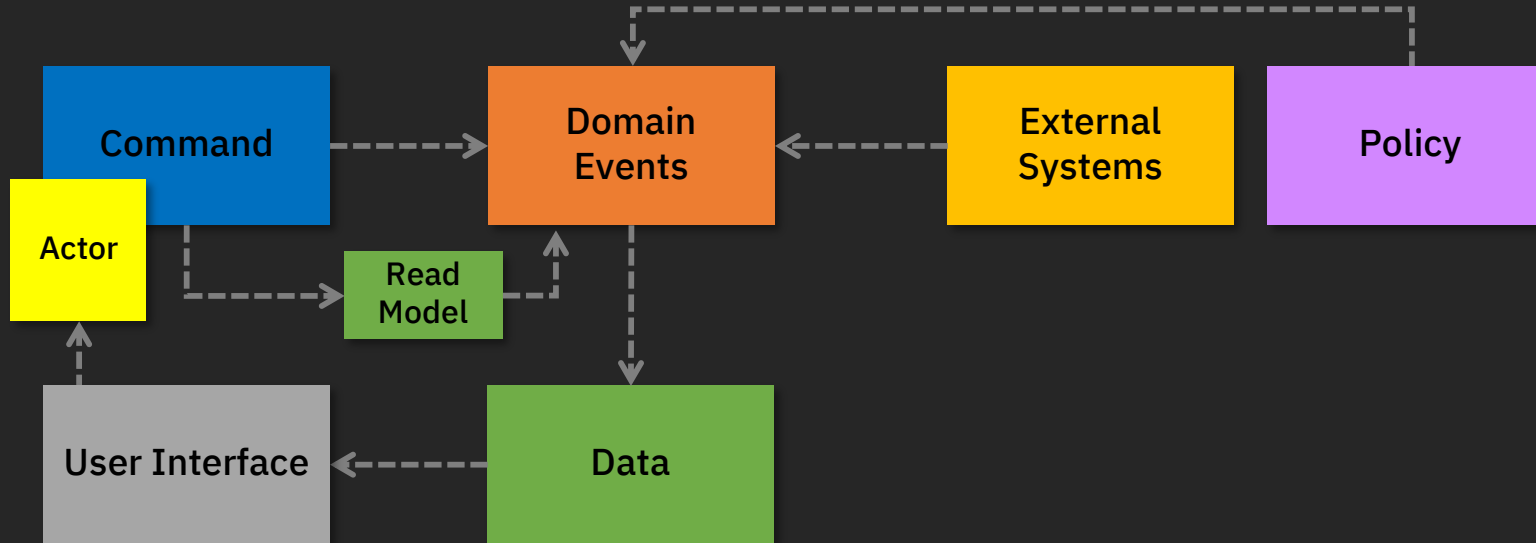
마이크로 서비스 설계

- UML 모델링 기법을 적용하여 마이크로 서비스를 식별함
- 사용자 스토리, 시퀀스 다이어그램, 클래스 다이어그램 등을 통해 상세 설계를 진행함
- 구현 계획을 수립함

바운디드 컨텍스트를 찾기 위한 방법 – Event Storming



Event Storming

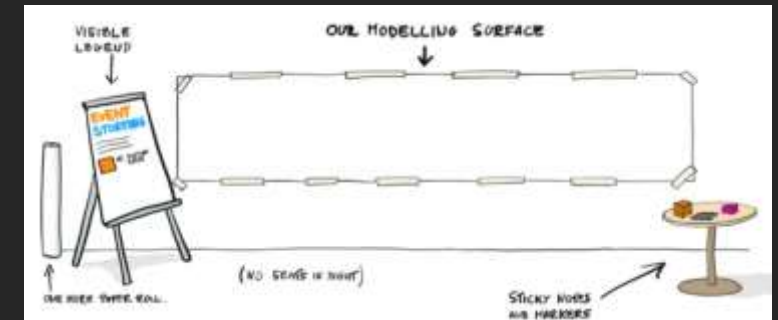


2019/2020년 IBM Garage를 금융/통신 등 다양한 수행 경험을 통해서

- DDD (Domain Driven Design)와 Event Storming, 마이크로 서비스 아키텍처에 대한 이해 및 경험을 통해 필수 역량을 내재화하고 있음
- IBM Garage Method에 대한 활용

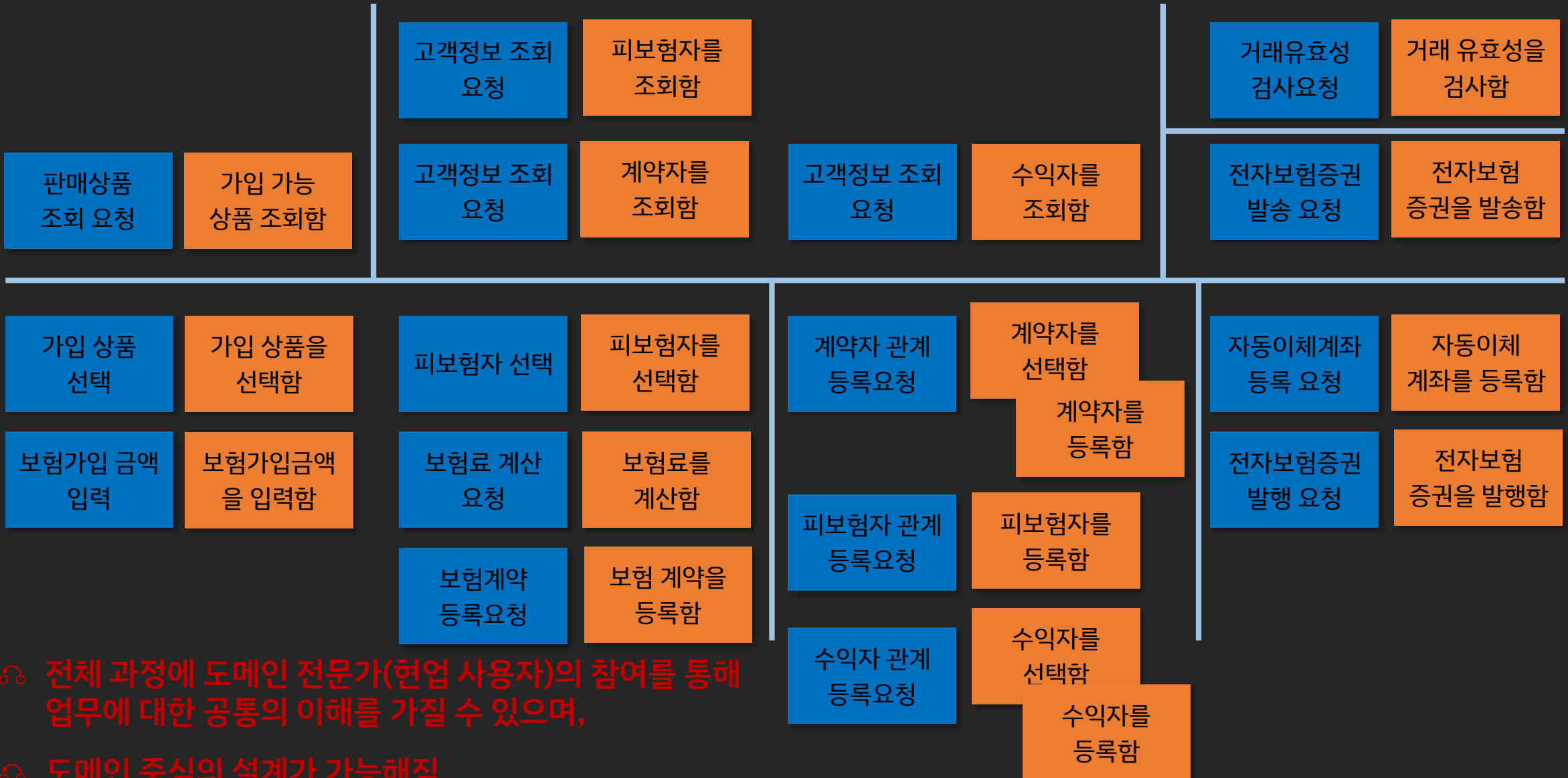
업무 도메인 중심의 설계를 성공적으로 수행하기 위해서는

- 비즈니스 및 IT 전문가의 협업을 통해서 (Event Storming 활용)
- 독립적인 비즈니스 도메인의 경계를 정의하고 (Bounded Context 정의)
- 바운디드 컨텍스트를 기준으로 팀을 지정하고 마이크로 서비스를 정의함



바운디드 컨텍스트 정의 – 이벤트/커맨드 정의

도메인: 보험 신계약 처리



⦿ 전체 과정에 도메인 전문가(현업 사용자)의 참여를 통해 업무에 대한 공통의 이해를 가질 수 있으며,

⦿ 도메인 중심의 설계가 가능해짐

바운디드 컨텍스트 정의 - 사용자/데이터 등 정의

도메인: 보험 신계약 처리

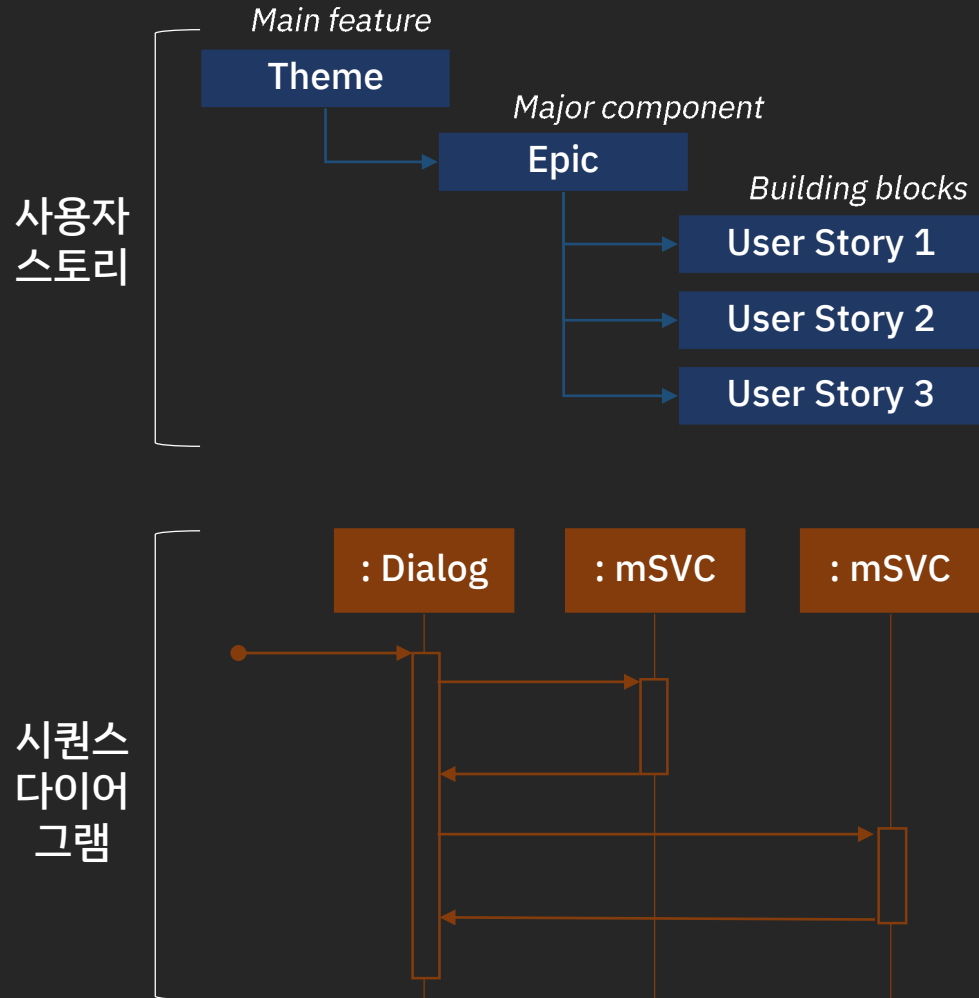


바운디드 컨텍스트 정의 - 바운디드 컨텍스트 정의

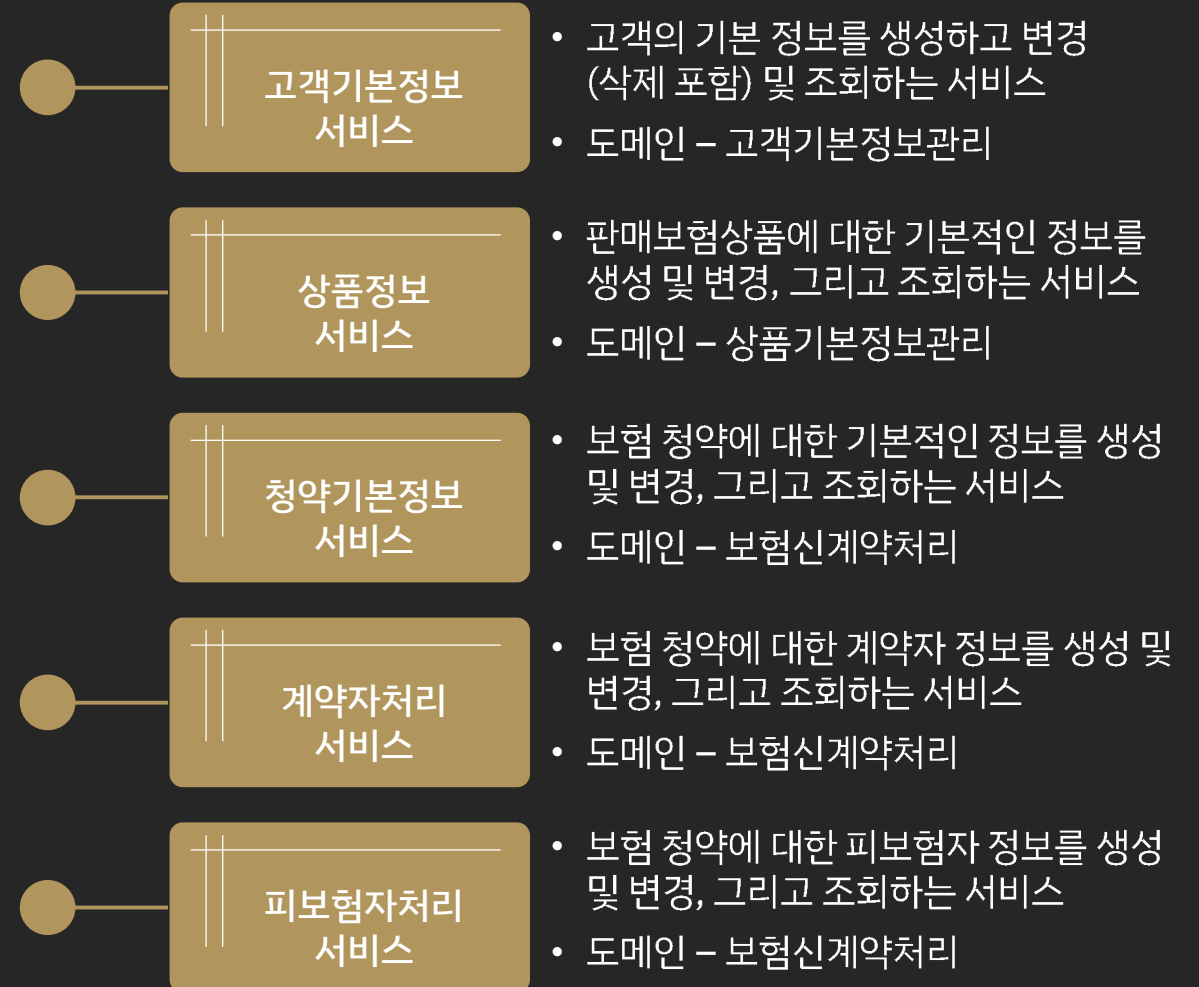
도메인: 보험 신계약 처리



마이크로 서비스 식별



마이크로 서비스 식별 예시



GUIDE 1

독립적이고 유연한 기능 중심의 마이크로 서비스를 정의하고 독립적으로 개발 및 배포할 수 있는 서비스 식별 및 설계에 대한 구체적인 방안 및 방법론

METHOD: 마이크로 서비스를 어떻게 식별 및 설계 할 것인가?

GUIDE 2

마이크로 서비스 단위로 데이터를 분리하기 위한 데이터 모델 구성 방안 및 서비스의 특성에 따른 데이터 솔루션 (NoSQL)의 적용 방안 및 가이드

DATA: 데이터 모델을 어떻게 정의하고 적용 솔루션은 ?

GUIDE 3

마이크로 서비스의 개발 및 운영 증가에 따라서 개발 및 운영의 효율성을 보장하고 지속하기 위해서 필요한 프로세스 및 조직 등에 대한 방안

DEVOPS: 어플리케이션을 어떻게 개발 및 운영할 것인가?

GUIDE 4

마이크로 서비스 아키텍처를 구현 및 배포/운영하기 위한 클라우드 플랫폼에 대한 아키텍처 및 고려사항

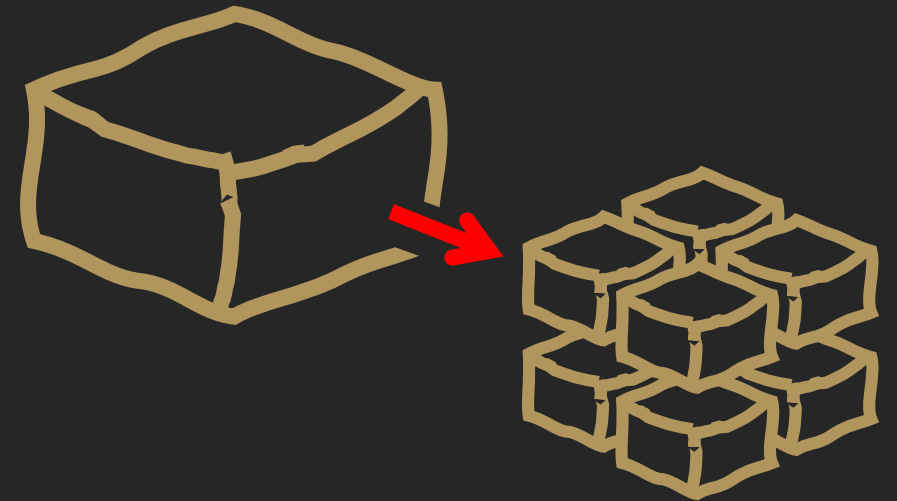
PLATFORM: 어플리케이션을 어디에 배포 및 운영할 것인가?

GUIDE 5

마이크로 서비스 아키텍처 및 클라우드 플랫폼에 대한 필수 역량을 어떻게 준비하고 지속적으로 적용하기 위해서 필요한 역량 내재화 방안

CAPABILITY: 지속적인 개발 및 운영을 위한 필요 역량 및 주체는 ?

- 🔗 Separate Data Model (데이터 모델 분리)
- 🔗 NoSQL (특성에 따른 솔루션 적용)



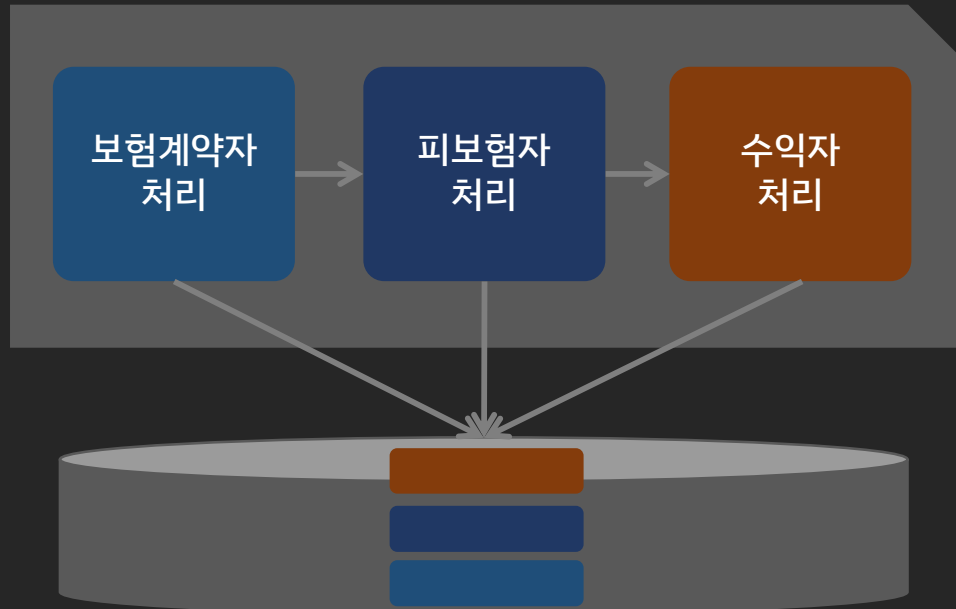
데이터 분리 예시 - 보험 당사자 처리 방안 (1/2)



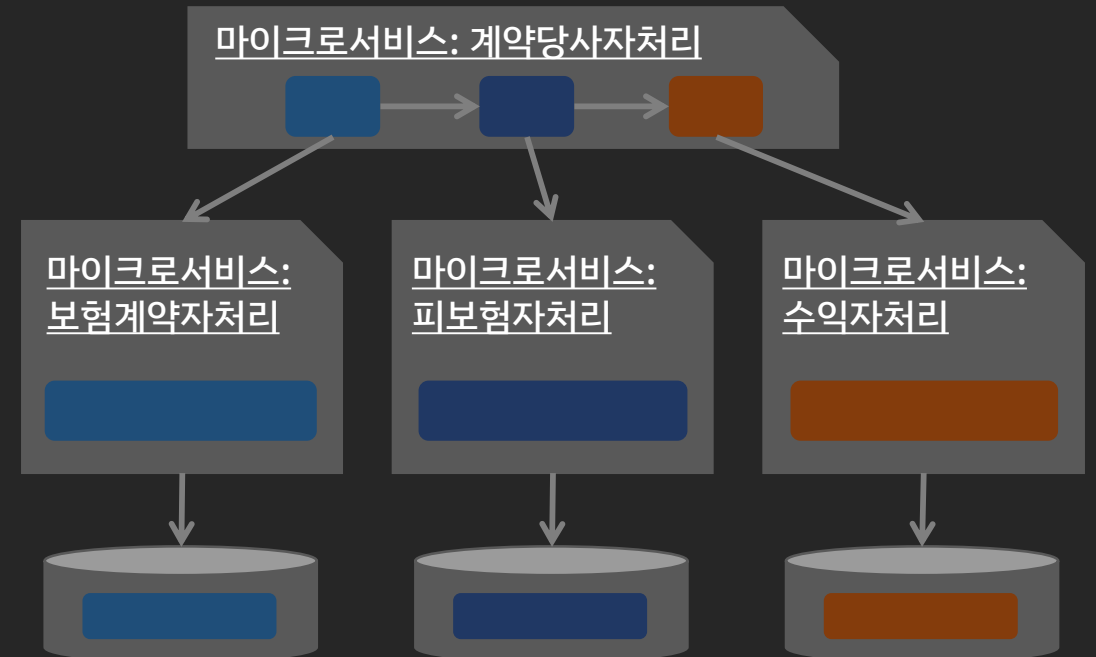
도메인 처리 조건

- ✓ 전제조건: 보험 신계약 정보와 개인 고객의 등록은 이미 처리가 완료됨
- ✓ 업무 기능: 계약당사자 어플리케이션은 생성/변경/조회/삭제 기능을 가지고 있으며, 계약자/피보험자/수익자에 대해서 처리함
- ✓ 비즈니스 규칙:
 - (1) 보험계약자는 1명을 지정할 수 있으며, 피보험자/수익자와 중복될 수 있음
 - (2) 피보험자와 수익자는 다수를 지정할 수 있음

모놀리틱 어플리케이션: 계약당사자처리



마이크로서비스: 계약당사자처리



데이터 분리 예시 - 보험 당사자 처리 방안 (2/2)

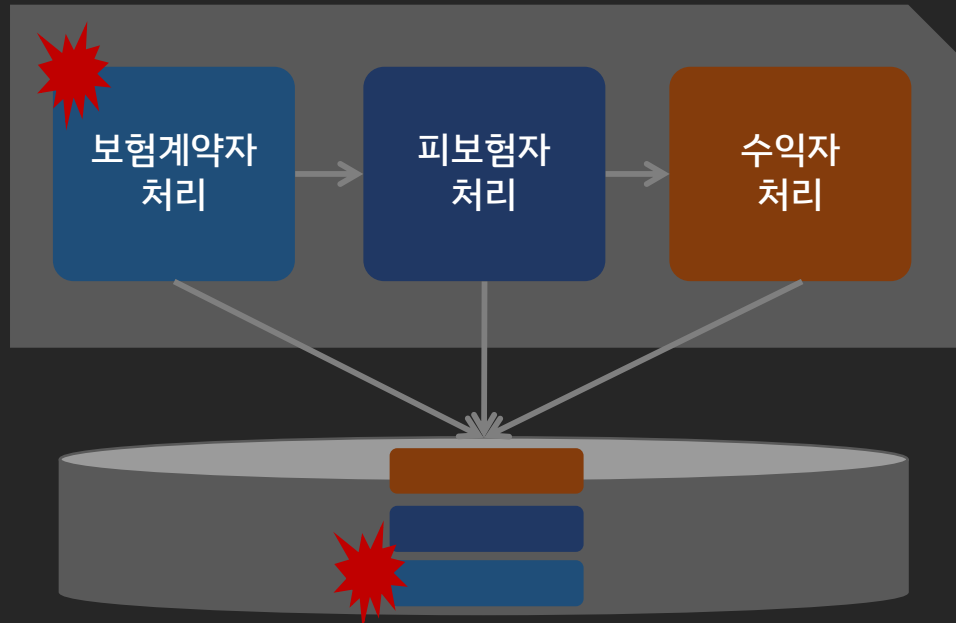


도메인 처리 조건

- ✓ 전제조건: 보험 신계약 정보와 개인 고객의 등록은 이미 처리가 완료됨
- ✓ 업무 기능: 계약당사자 어플리케이션은 생성/변경/조회/삭제 기능을 가지고 있으며, 계약자/피보험자/수익자에 대해서 처리함
- ✓ 비즈니스 규칙:
 - (1) (변경)보험계약자는 다수를 등록할 수 있으며, 대표계약자, 보험료납입 비율을 지정해야 함
 - (2) 피보험자와 수익자는 다수를 지정할 수 있음

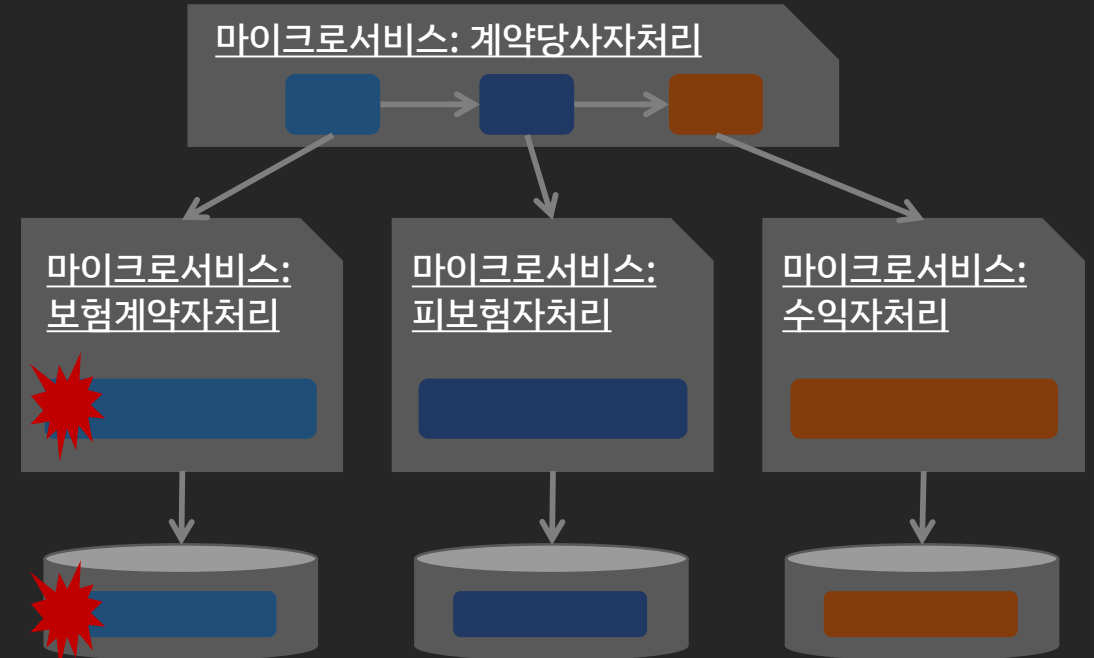
모놀리틱 어플리케이션: 계약당사자처리

“변경이 어플리케이션/데이터 모두에 발생함”



마이크로서비스: 계약당사자처리

“변경이 보험계약자처리 서비스에만 발생하며 빠르게 적용 가능함”



GUIDE 1

독립적이고 유연한 기능 중심의 마이크로 서비스를 정의하고 독립적으로 개발 및 배포할 수 있는 서비스 식별 및 설계에 대한 구체적인 방안 및 방법론

METHOD: 마이크로 서비스를 어떻게 식별 및 설계 할 것인가?

GUIDE 2

마이크로 서비스 단위로 데이터를 분리하기 위한 데이터 모델 구성 방안 및 서비스의 특성에 따른 데이터 솔루션 (NOSQL)의 적용 방안 및 가이드

DATA: 데이터 모델을 어떻게 정의하고 적용 솔루션은?

GUIDE 3

마이크로 서비스의 개발 및 운영 증가에 따라서 개발 및 운영의 효율성을 보장하고 지속하기 위해서 필요한 프로세스 및 조직 등에 대한 방안

DEVOPS: 어플리케이션을 어떻게 개발 및 운영할 것인가?

GUIDE 4

마이크로 서비스 아키텍처를 구현 및 배포/운영하기 위한 클라우드 플랫폼에 대한 아키텍처 및 고려사항

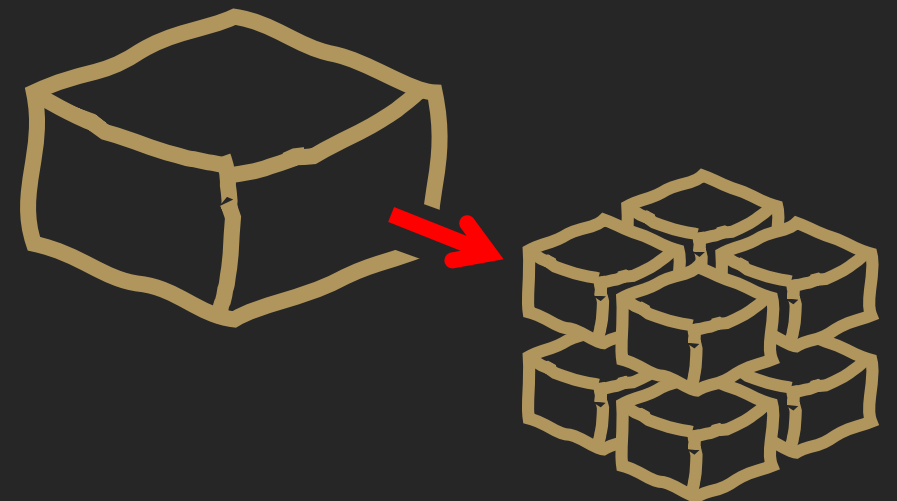
PLATFORM: 어플리케이션을 어디에 배포 및 운영할 것인가?

GUIDE 5

마이크로 서비스 아키텍처 및 클라우드 플랫폼에 대한 필수 역량을 어떻게 준비하고 지속적으로 적용하기 위해서 필요한 역량 내재화 방안

CAPABILITY: 지속적인 개발 및 운영을 위한 필요 역량 및 주체는?

🔗 DevOps (BizDevOps, DevSecOps)
(일하는 방식의 변화를 기반으로 계획 및 실행)



DevOps (데브옵스)에 대한 이해



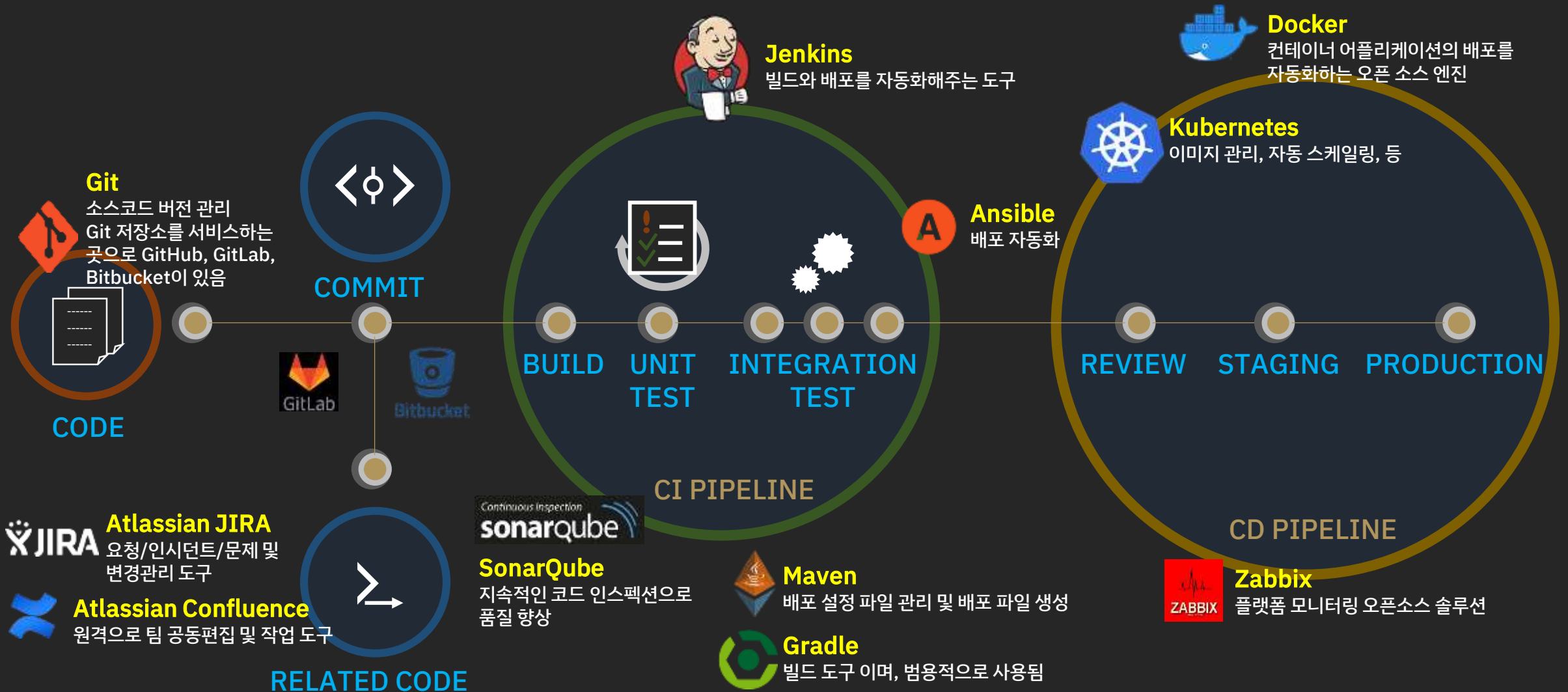
데브옵스는 애플리케이션과 서비스를 빠른 속도로 제공할 수 있도록 조직의 역량을 향상시키는 **문화 철학, 방식 및 도구의 조합**이며,
 기존의 소프트웨어 개발 및 인프라 관리 프로세스를 사용하는 조직보다 제품을 더 빠르게 혁신하고 개선할 수 있음
 이러한 빠른 속도를 통해 조직은 고객을 더 잘 지원하고 시장에서 좀 더 효과적으로 경쟁할 수 있음



주요 원칙 (CALMS)

Culture	Automation	Lean	Measurement	Share
<p>문화(Culture) DevOps 문화는 한 마디로 협업임. 이 세상의 모든 도구와 자동화 시스템은 Dev와 Ops의 협력하고자 하는 마음이 없다면 어떤 것도 해결할 수 없음</p>	<p>자동화(Automation) 자동화는 위에서 말했듯이 개발, 테스트 및 지속적 배포의 핵심요소임. 개발 사이클 상에서 높은 수준의 자동화를 도입한다면 막대한 이익을 얻을 수 있음</p>	<p>간소화(Lean) 낭비를 제거함으로써 어떻게 고객에게 가치를 빠르게 제공할 수 있을까에 대한 생각이자 사고방식 DevOps 상에서는 이러한 낮은 가치를 지닌 것을 끊임없이 찾고 지속적으로 개선하는 것임</p>	<p>측정(Measurement) DevOps는 여러 가지를 측정하고, 측정 결과를 가시화하여 보여줌. 실질적인 데이터를 보여줌으로써 지속적인 개선을 위한 노력이, 실제 개선으로 이어지고 있는지 확인할 수 있음</p>	<p>공유(Share) 성공 여부와 상관없이 DevOps는 다른 사람들이 배울 수 있도록 서로의 경험을 공유함. Dev와 Ops 간의 마찰은 주로 공유의 부족으로 인해 발생하게 됨</p>

DevOps with CI/CD (Continuous Integration/Continuous Delivery)



GUIDE 1 독립적이고 유연한 기능 중심의 마이크로 서비스를 정의하고 독립적으로 개발 및 배포할 수 있는 서비스 식별 및 설계에 대한 구체적인 방안 및 방법론

METHOD: 마이크로 서비스를 어떻게 식별 및 설계 할 것인가?

GUIDE 2 마이크로 서비스 단위로 데이터를 분리하기 위한 데이터 모델 구성 방안 및 서비스의 특성에 따른 데이터 솔루션 (NOSQL)의 적용 방안 및 가이드

DATA: 데이터 모델을 어떻게 정의하고 적용 솔루션은 ?

GUIDE 3 마이크로 서비스의 개발 및 운영 증가에 따라서 개발 및 운영의 효율성을 보장하고 지속하기 위해서 필요한 프로세스 및 조직 등에 대한 방안

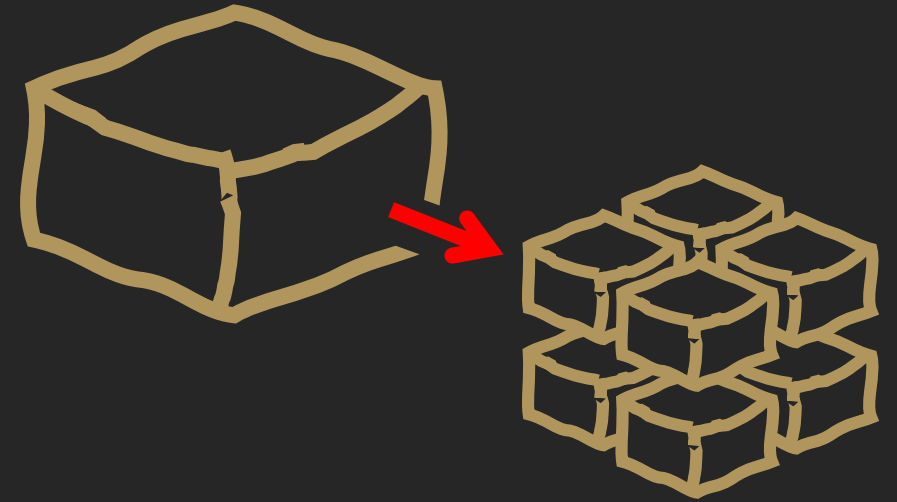
DEVOPS: 어플리케이션을 어떻게 개발 및 운영할 것인가?

GUIDE 4 마이크로 서비스 아키텍처를 구현 및 배포/운영하기 위한 클라우드 플랫폼에 대한 아키텍처 및 고려사항

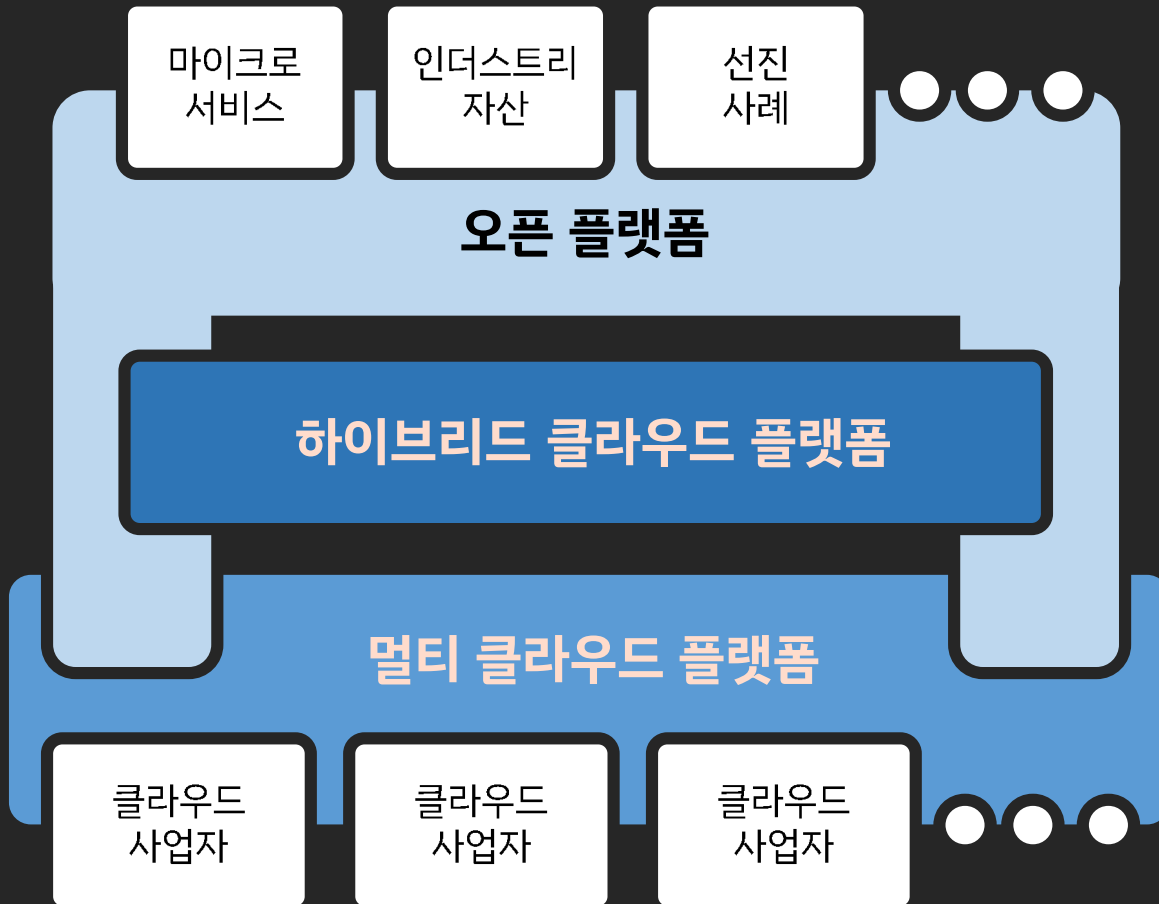
PLATFORM: 어플리케이션을 어디에 배포 및 운영할 것인가?

GUIDE 5 마이크로 서비스 아키텍처 및 클라우드 플랫폼에 대한 필수 역량을 어떻게 준비하고 지속적으로 적용하기 위해서 필요한 역량 내재화 방안

CAPABILITY: 지속적인 개발 및 운영을 위한 필요 역량 및 주체는 ?



Open Hybrid Multi-Cloud Platform
(오픈 하이브리드 멀티 클라우드 계획 및 적용)



오픈 하이브리드 멀티 클라우드 플랫폼은

- 독립적이며 의미를 갖는 최소 단위의 비즈니스 역량 (마이크로 서비스)의 실행 및 연계를 지원하며,
- 다양한 인더스트리 솔루션 및 선진 사례, 검증된 서비스를 쉽게 수용할 수 있으며,
- 대체 가능한 서비스 혹은 솔루션을 유연하고 빠르게 변경 적용이 가능하며,
- 온프레미스와 프라이빗, 퍼블릭 클라우드의 유연한 구성 및 확장을 가능하게 하고,
- 다양한 클라우드 사업자를 쉽게 통합할 수 있는 플랫폼 환경을 제공해야 함

멀티 클라우드 플랫폼 적용 사례



- ☞ **Pain Point 1.**
퍼블릭 환경과 프라이빗 환경의 통합 모니터링 요건
- ☞ **Pain Point 2.**
통합 플랫폼은 안전하게 온-프레미스 환경에 구축
- ☞ **Pain Point 3.**
VM 기반과 Kubernetes 환경을 통합 관리할 수 있는 플랫폼 필요 (확장성 고려)



- ☞ 모듈식으로 기능 추가 가능
- ☞ 설치형으로 온프레미스 환경에 구축
- ☞ 하이브리드 멀티 클라우드 관리를 위한 Full Set 보유

GUIDE 1

독립적이고 유연한 기능 중심의 마이크로 서비스를 정의하고 독립적으로 개발 및 배포할 수 있는 서비스 식별 및 설계에 대한 구체적인 방안 및 방법론

METHOD: 마이크로 서비스를 어떻게 식별 및 설계 할 것인가?

GUIDE 2

마이크로 서비스 단위로 데이터를 분리하기 위한 데이터 모델 구성 방안 및 서비스의 특성에 따른 데이터 솔루션 (NOSQL)의 적용 방안 및 가이드

DATA: 데이터 모델을 어떻게 정의하고 적용 솔루션은?

GUIDE 3

마이크로 서비스의 개발 및 운영 증가에 따라서 개발 및 운영의 효율성을 보장하고 지속하기 위해서 필요한 프로세스 및 조직 등에 대한 방안

DEVOPS: 어플리케이션을 어떻게 개발 및 운영할 것인가?

GUIDE 4

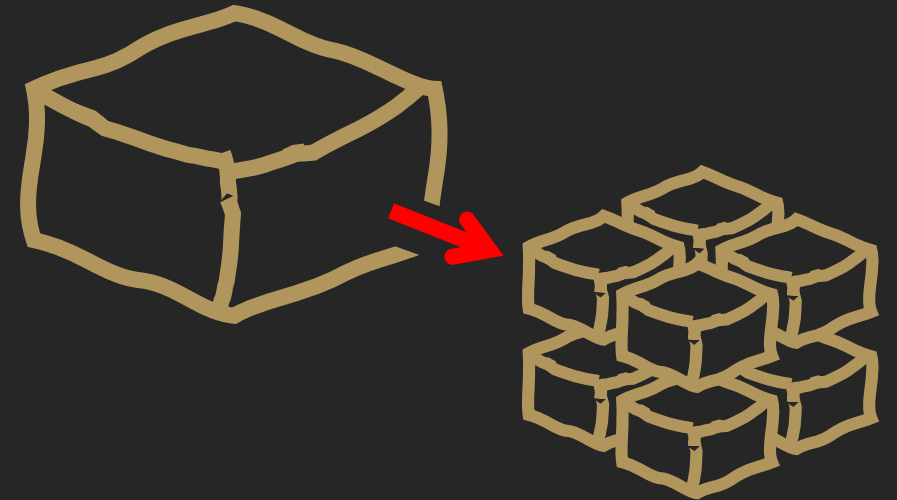
마이크로 서비스 아키텍처를 구현 및 배포/운영하기 위한 클라우드 플랫폼에 대한 아키텍처 및 고려사항

PLATFORM: 어플리케이션을 어디에 배포 및 운영할 것인가?

GUIDE 5

마이크로 서비스 아키텍처 및 클라우드 플랫폼에 대한 필수 역량을 어떻게 준비하고 지속적으로 적용하기 위해서 필요한 역량 내재화 방안

CAPABILITY: 지속적인 개발 및 운영을 위한 필요 역량 및 주체는?



Leading Practice
(역량 내재화를 통한 주도적 변화)

역량 내재화를 통한 주도적 변화



- ☞ 도메인 전문가 (현업 사용자)의 참여로 설계 및 개발을 하고 있으며,
- ☞ 설계 및 개발/운영의 필요 역량에 대한 내재화를 함께 진행하고 있음

Discover:

클라우드 전환은 단순히 인프라의 변경 혹은 기술의 전환이 아니라 비즈니스에 대한 **“문제 인식 및 개선 아이디어”** 적용이 수반되어야 함

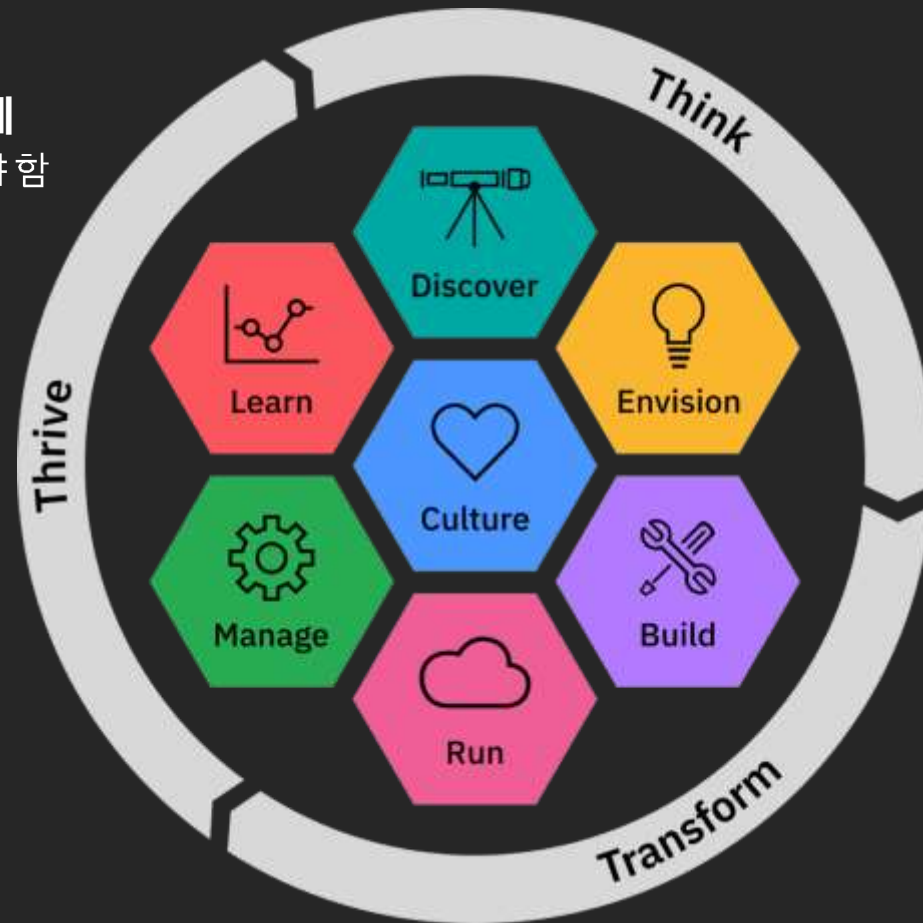
Learn:

개발 및 운영하고 있는 서비스에 대해서 지속적으로 모니터링 및 피드백을 수집하고 **“지속적으로 반영”**할 수 있어야 함

Manage:

클라우드 기술에 대한 역량 내재화를 통해서 **“서비스에 대한 안정적 운영”**을 할 수 있어야 함

Full Lifecycle of MSA Adoption



Culture:

지속적으로 클라우드 전환을 성공적으로 수행하기 위해서는 반드시 **“일하는 방식의 변화”**가 수반되어야 함

Envision:

어플리케이션 현대화 및 클라우드 네이티브 이해 및 경험을 통해서 **“아이디어 및 아키텍처 구체화”**를 진행함

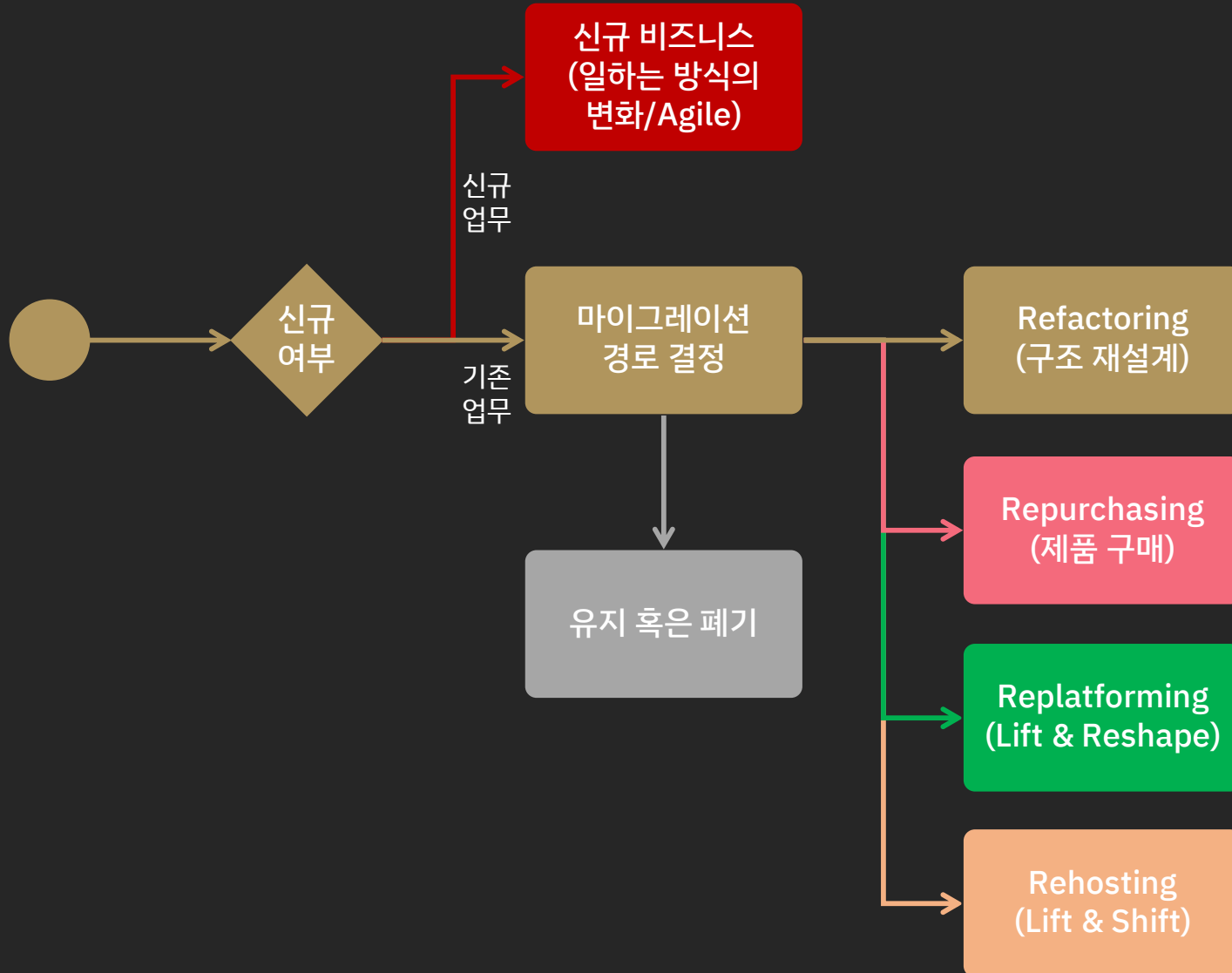
Build:

클라우드 기술에 대한 이해 및 내재화를 통해서 **“어플리케이션 및 서비스에 대한 구현”**이 가능해야 함

Run:

클라우드 플랫폼 기술에 대한 이해 및 내재화를 통해 **“어플리케이션 및 서비스에 대한 배포 및 수행”**이 가능해야 함

- 사례 이해 및 마이크로 서비스 필요성
- 사례로 이해하는 마이크로 서비스 접근 방안
- 활용 가능한 오픈링 및 도구, 그리고 권고사항



IBM Garage

최근 금융 및 통신사 등 8개의 사례를 통해 경험 및 역량을 보유하고 있으며, 경쟁사와 차별화된 요소임

App Modernization Service

Mono to Micro & Cloud Paks

최근 새로운 서비스 출시를 계획하는 금융권의 마이크로 서비스 설계 및 클라우드 구현을 위한 기술 검증, 아키텍처 수립 등

Transformation Advisor (TA) & Cloud Paks

Transformation Advisor (TA) & Cloud Paks

역량 내재화를 통한 서비스의 빠른 적용 – IBM Garage Service



- 기간: 2시간
- 사전 워크숍을 통해서 블록체인 및 클라우드 적용 대상 비즈니스 혹은 유스케이스에 대한 이해 및 방향 결정
- 업무/유스케이스 사전 이해를 통한 사전 준비 내용 확인 및 목표에 대한 명확한 공유
- 업무 담당 및 개발 담당 직원 필참

- 기간: 5주 ~ 8주 (유스케이스 복잡도에 따라 결정)
- Framing workshop에서 결정된 비즈니스 혹은 유스케이스에 대한 문제점 식별 및 구체화, 개선을 위한 다양한 아이디어 발굴 및 우선 순위 정의
- MVP 우선 적용 대상 지정 및 주요 목표와 효과에 대한 정의 및 이해
- 디지털 신기술 기반의 구현을 위한 아키텍처 요구 확인 및 아키텍처 구조에 대한 설계
- 클라우드 네이티브 적용을 위한 아키텍처 구성 요소에 대한 확인 및 정의
- 블록체인 및 클라우드 기술을 통한 서비스의 개발/검증 그리고 배포



- 애자일 방식의 경험을 통한 내재화
- 개발 과정에서 다양한 애자일 프랙티스에 대한 직접적인 경험 및 내재화
 - 애자일 방식의 적용을 위한 가이드 및 피드백

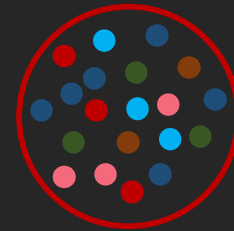
- MVP 이후 계획에 따라 해당 유스케이스의 지속적인 확대 개발 및 적용/운영을 위한 확산 개발
- 개발 및 운영을 통한 피드백의 반영
- 비즈니스 성과 및 효과에 대한 점검 및 확인

마이크로 서비스 식별을 권고하는 도구 - Mono to Micro (M2M)

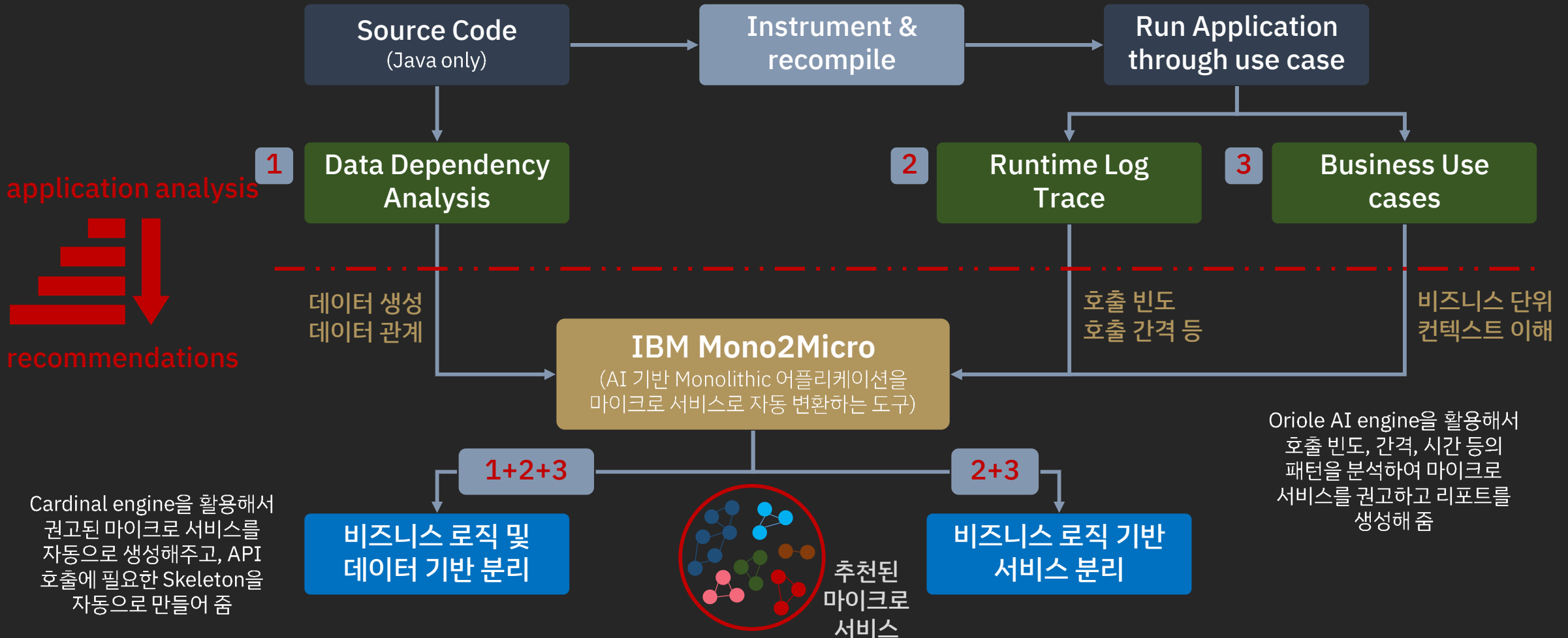


빠른 의사결정 및 전환

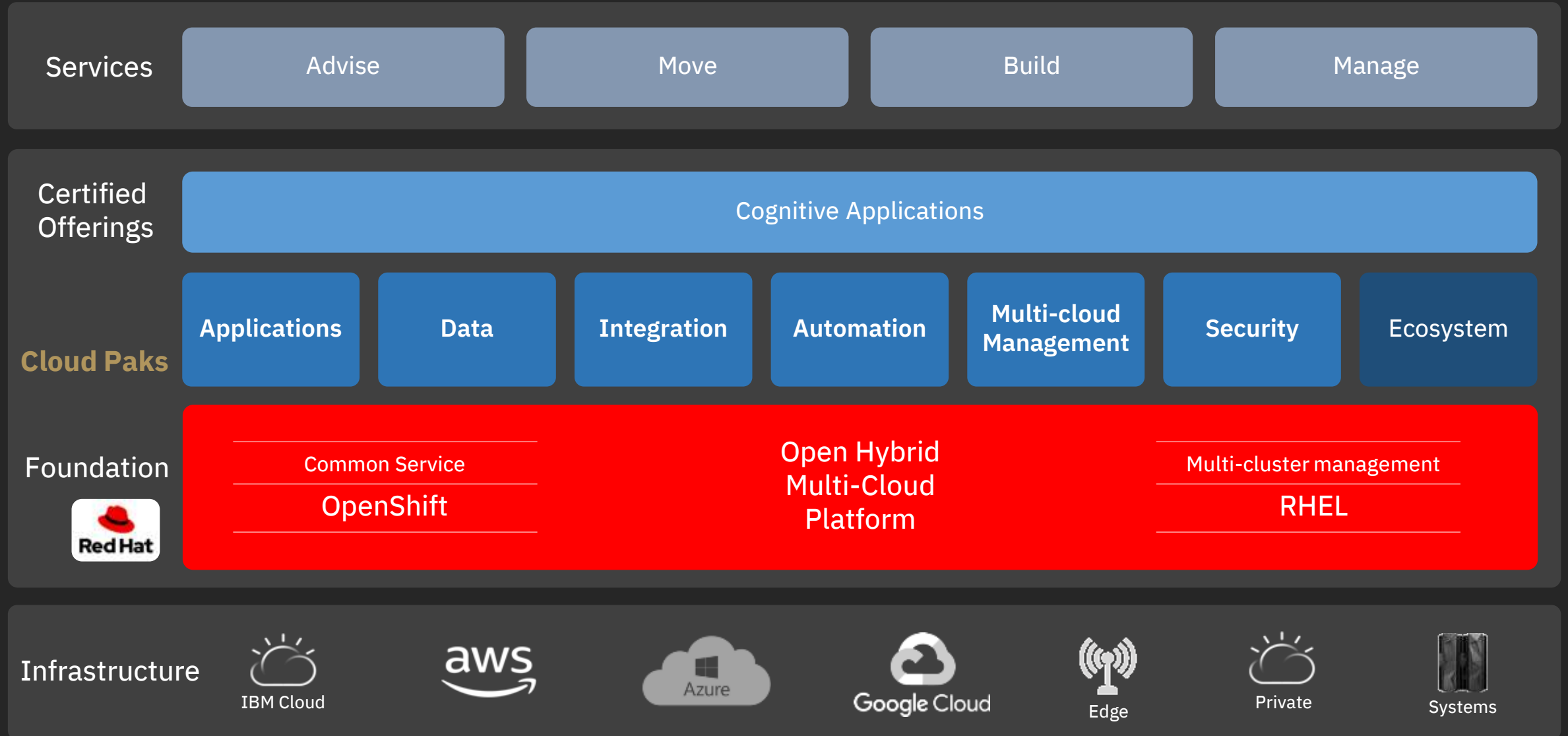
리스크 감소 및 비용 절감



Monolithic
어플리케이션



오픈 하이브리드 멀티 클라우드 플랫폼 – IBM Cloud Paks – Build Once. Deploy Anywhere.





1 현행 클라우드 전략에 대한 검토

클라우드 전략을 오픈 하이브리드 멀티 클라우드 관점에서 검토하고 부족한 부분에 대한 보완이 필요함 (클라우드 전략이 명확하지 않은 경우 Blueprint를 정의할 필요 있음)

2 새로운 서비스에 대한 클라우드 적용 우선 고려

새로운 서비스를 출시하는 경우에 클라우드 플랫폼 적용을 우선 고려하고 클라우드 네이티브 어플리케이션 적용으로 빠르고 경쟁력 있는 서비스에 대한 제공

3 다양한 디지털 경험을 제공하는 플랫폼 제시

클라우드 뿐만 아니라 데이터, 인공지능 등의 디지털 기술을 적용하고 경험할 수 있는 플랫폼 환경에 대한 우선 적용을 통해서 이해당사자 모두 쉽게 참여하고 피드백 할 수 있는 플랫폼 제시

4 역량 내재화를 통한 주도적 실행 필요

클라우드 기술 뿐만 아니라 Agile 등의 전반적인 역량에 대한 수준 진단 및 갭이 있는 필수 역량에 대해서 역량 내재화 방안을 정의하고 실행 계획에 따라 시작함

THANK YOU

임용성 상무

yslim@kr.ibm.com

IBM GARAGE PRACTICE LEADER

IBM CLOUD INTEGRATION EXPERT SERVICE LEADER