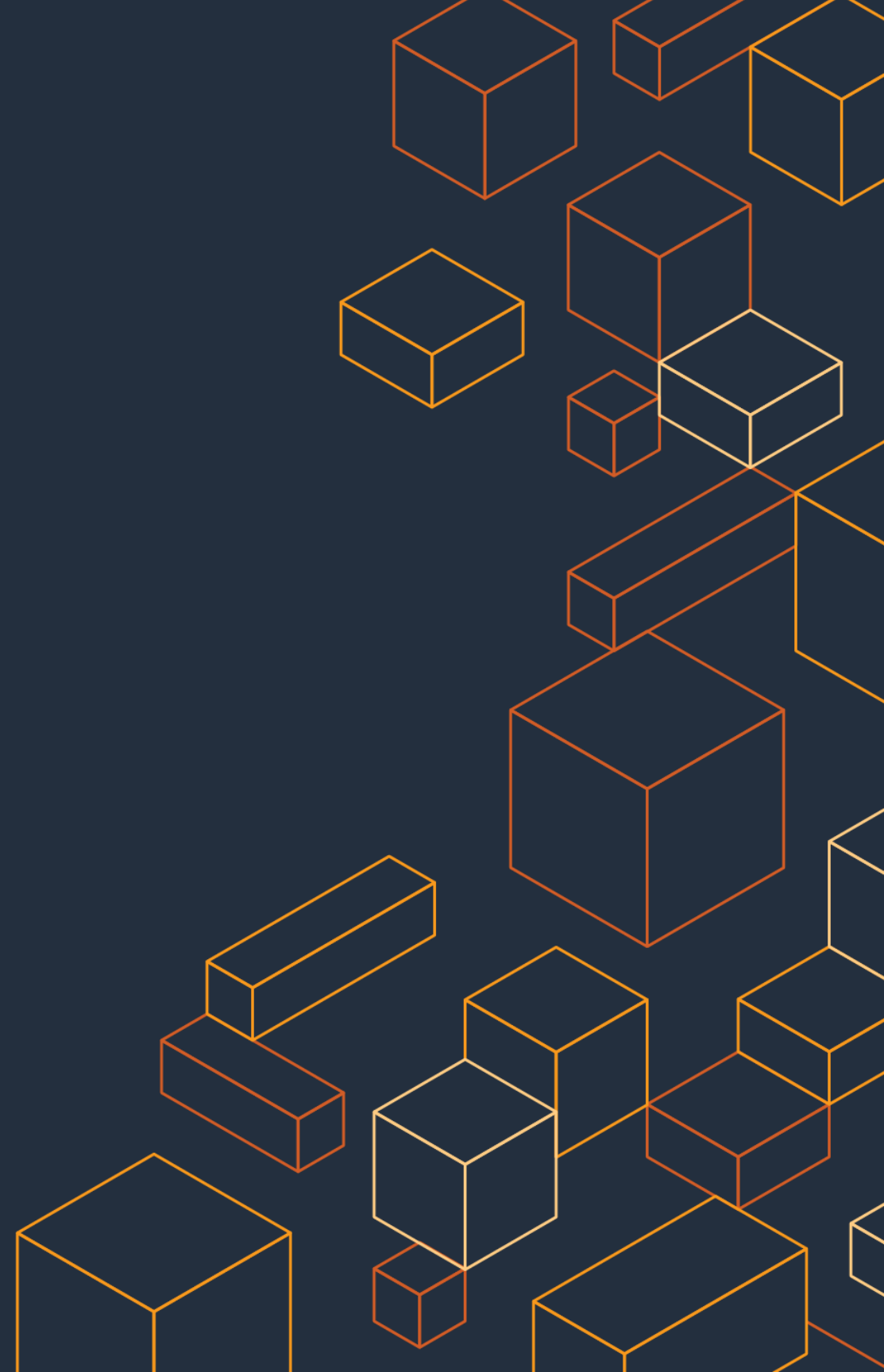




Database Modernization on AWS

Database Solution Architect
11.28.2021



Agenda

- What is a Modern Application ?
- Modern Application Architecture
- DBMS Modernization Guide

What is a Modern Application ?

IT trend

Mainframe



Client Server



Three tier



Microservices



What is a Modern Application ?

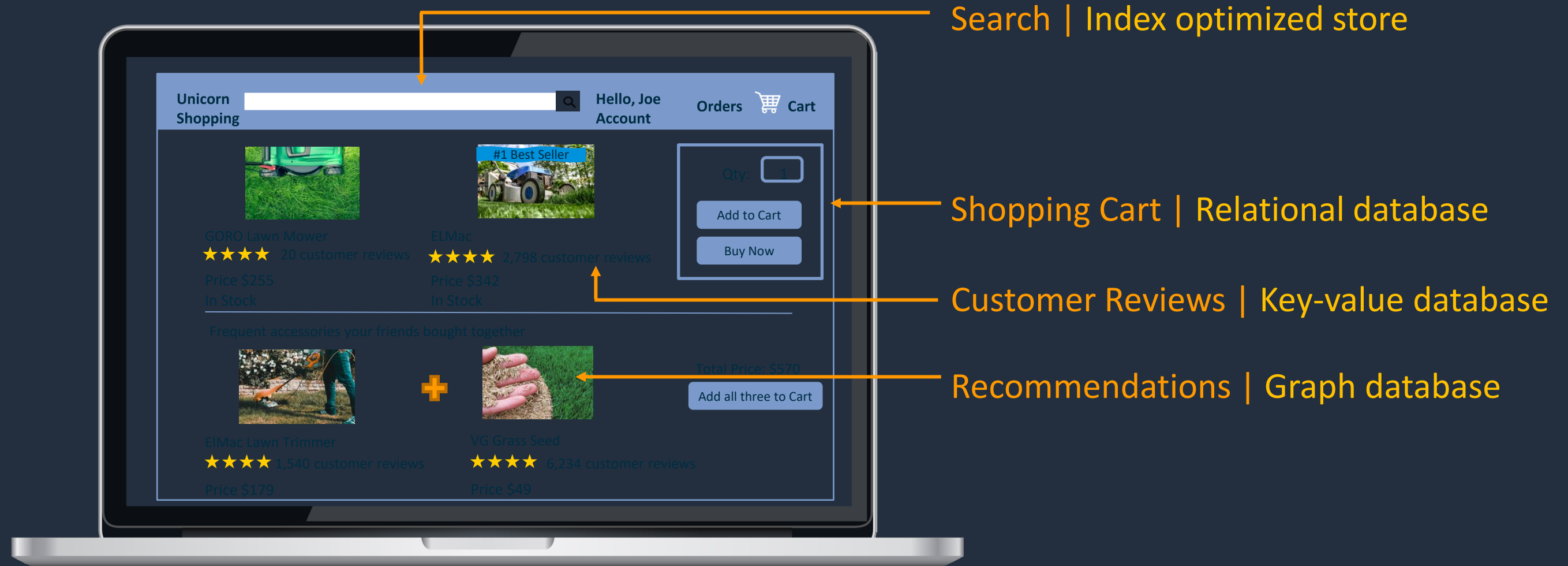
Modern Application Profile



Users	1M+
Data volume	Terabytes–petabytes
Locality	Global
Performance	Microsecond latency
Request rate	Millions per second
Access	Mobile, IoT, devices
Scale	Virtually unlimited
Economics	Pay-as-you-go
Developer access	Instance API access
Development	Apps and storage are decoupled

What is a Modern Application ?

Modern applications require a variety of data stores



Modern Application Architecture

Traditional Application Architecture



Web servers

Presentation layers



Application servers

Business logic

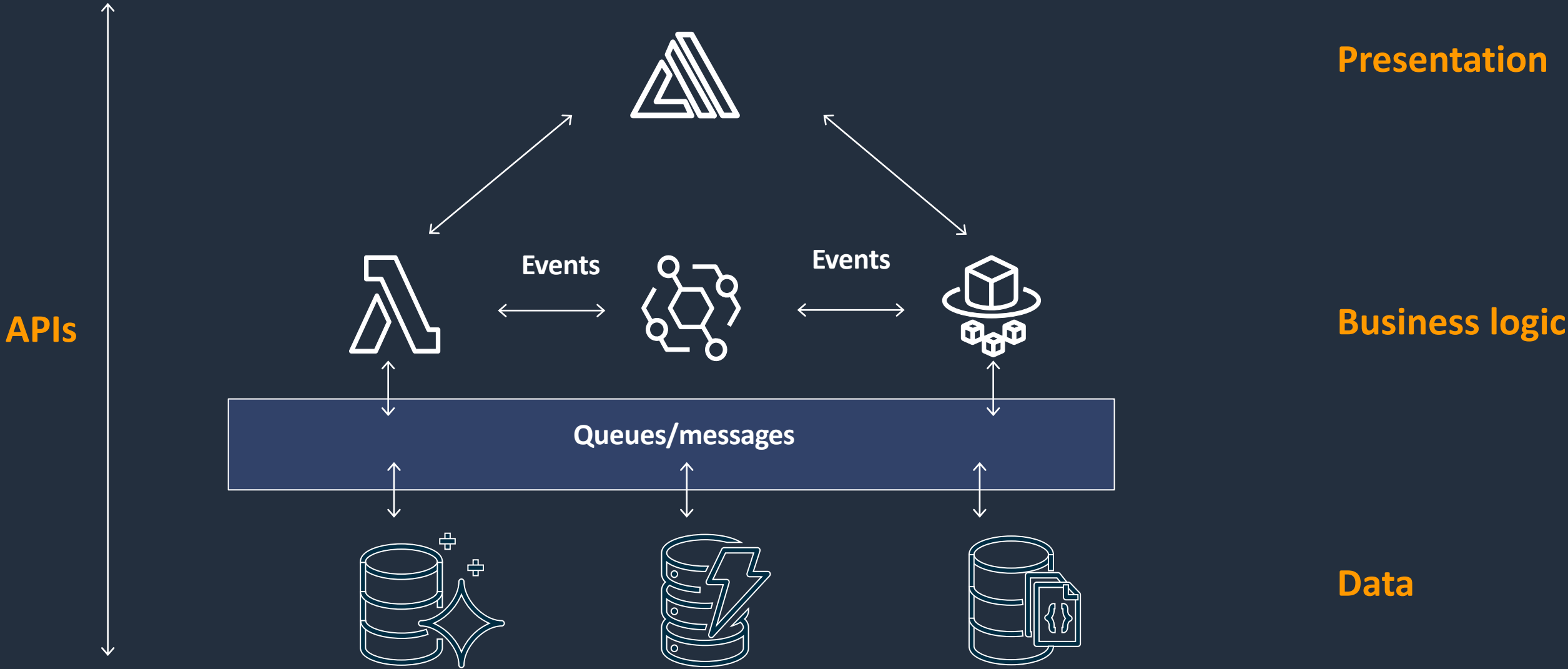


Database servers

Data layer

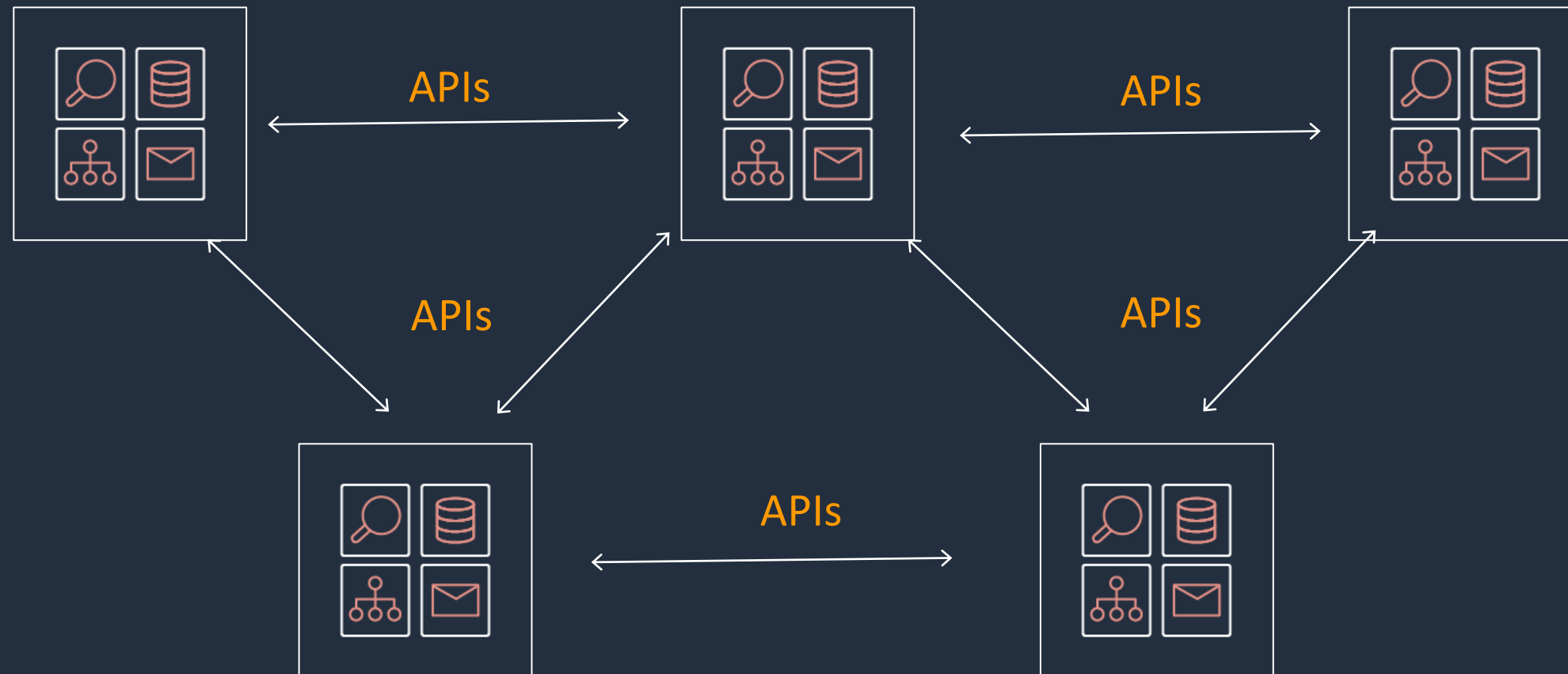
Modern Application Architecture

Modern Application Reference Architecture



Modern Application Architecture

Reference Architecture Zoom-out



Modern Application Architecture

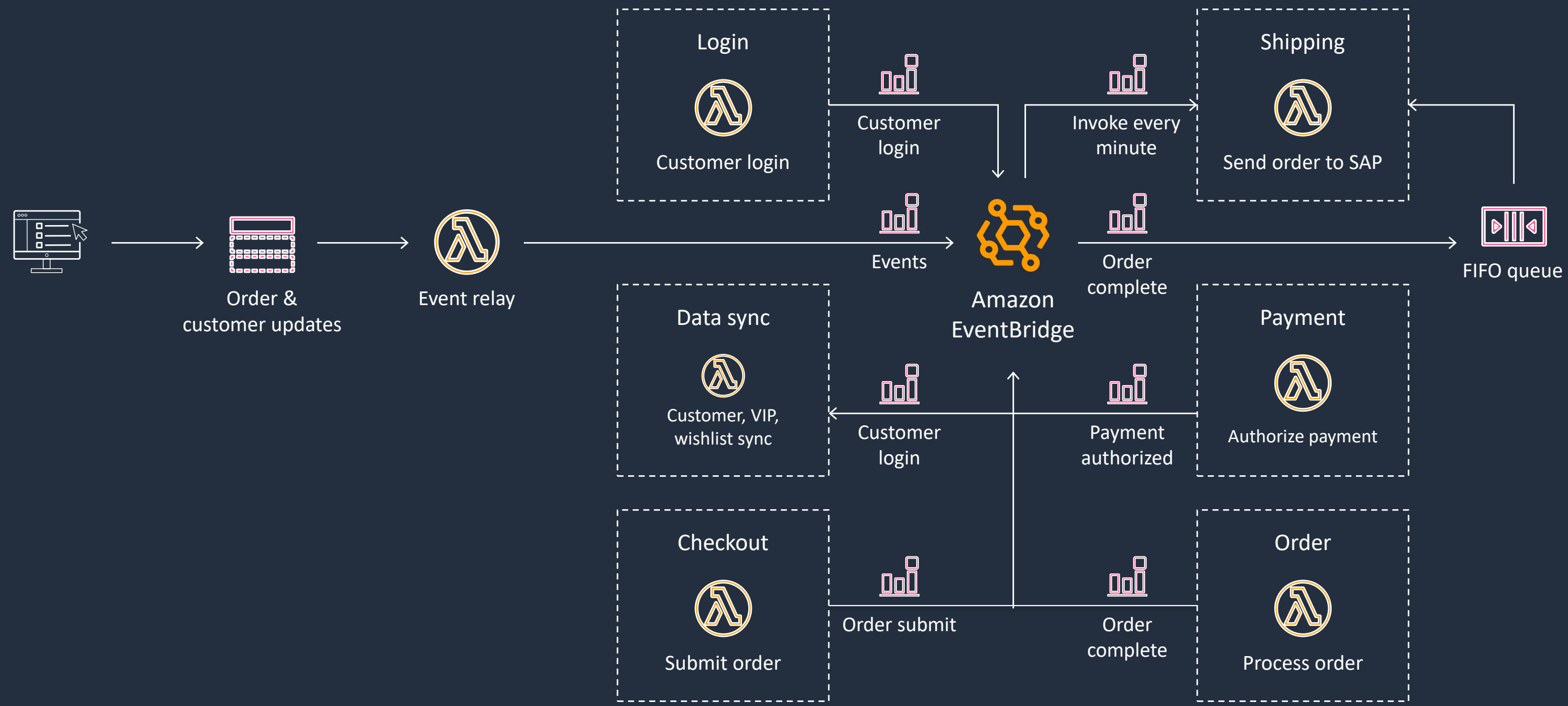
Integrate Applications & Modular Services

	동기식: API 기반	비동기식: EVENT 기반
Inter/intra-service	일반적인 앱 간 통신	일반적인 앱 내 통신
확장성	지점 간 연결을 관리하는 데 필요한 도구	거의 무한대로 확장 가능
비용	최대 사용을 위한 프로비저닝은 CPU 사용률을 낮춤	"사용한 만큼 지불"의 비용 이점
지연 시간	매우 낮을 수 있음	이론상 더 높지만 대기 시간 요구 사항은 생각만큼 낮지 않음
민첩성	시작하기 쉬움. 대규모로 Point-to-Point를 사용하기 어려움	분리된 시스템은 민첩성을 극적으로 증가시킴



Modern Application Architecture

Event-Driven Design



Modern Application Architecture

Purpose-built Databases



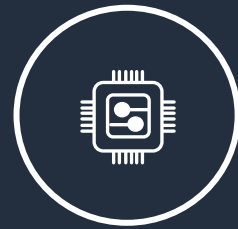
Relational



Key value



Document



In-memory



Graph



Time series



Ledger



Wide column



Aurora



RDS



DynamoDB



DocumentDB



ElastiCache



Neptune



Timestream



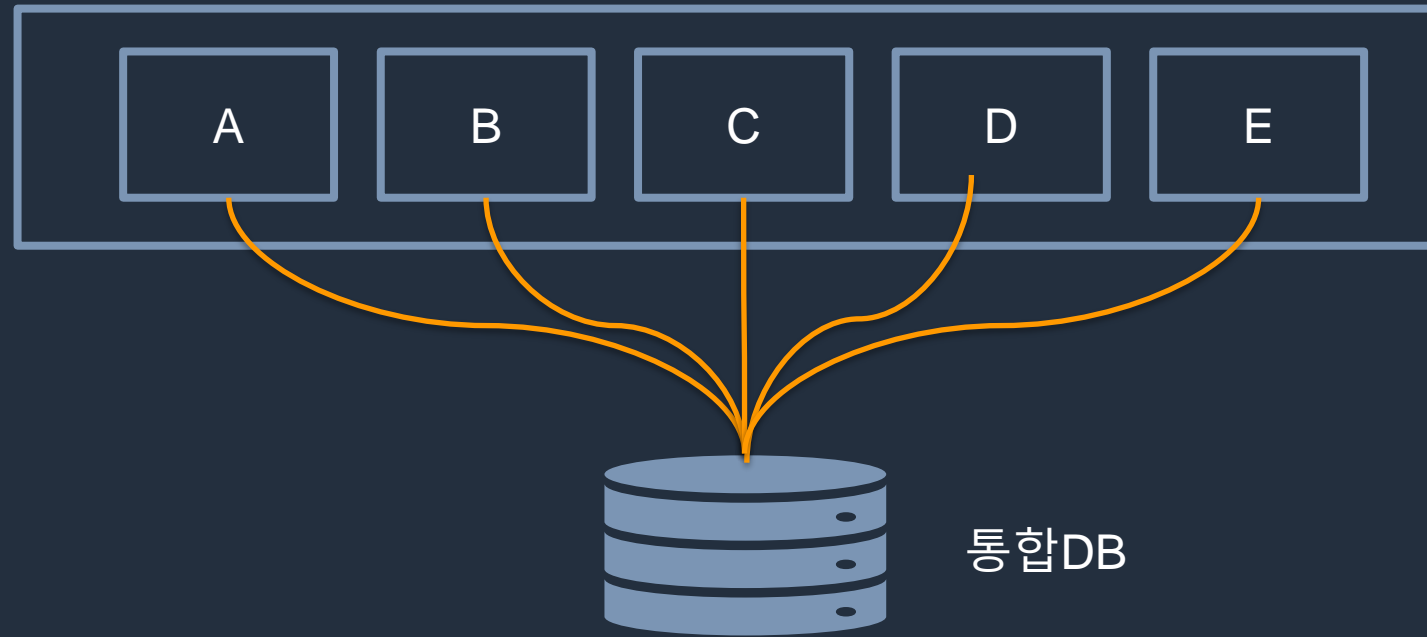
QLDB



Managed
Cassandra

DBMS Modernization Guide

Traditional DBMS Pains

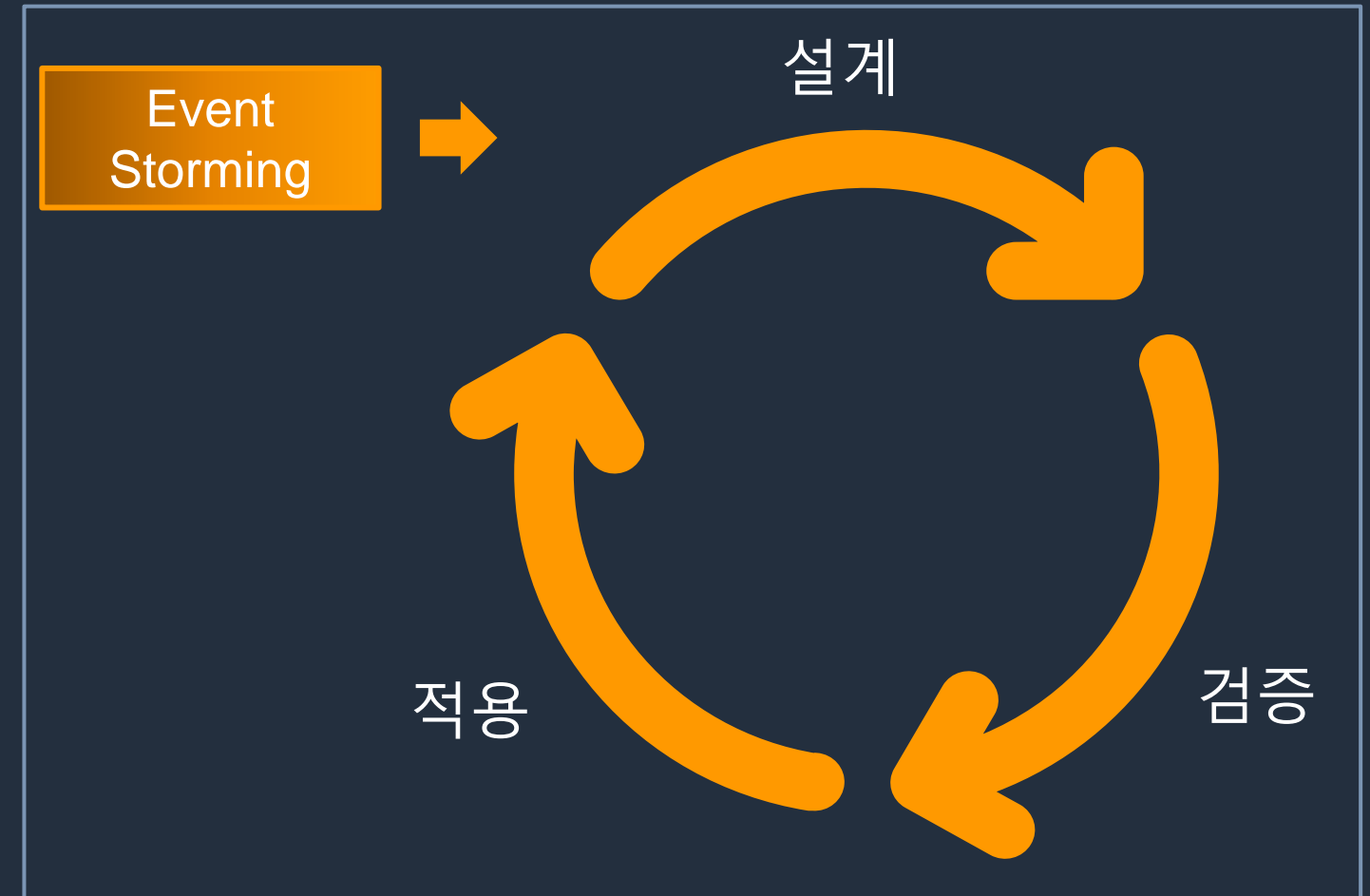


- 기능 별 모듈화 수준
- 데이터의 통합 (장애 전파에 취약)
- 기능 별 모듈화의 각 의존성 존재
- 전형적인 분산 모놀리스 아키텍처

DBMS Modernization Guide

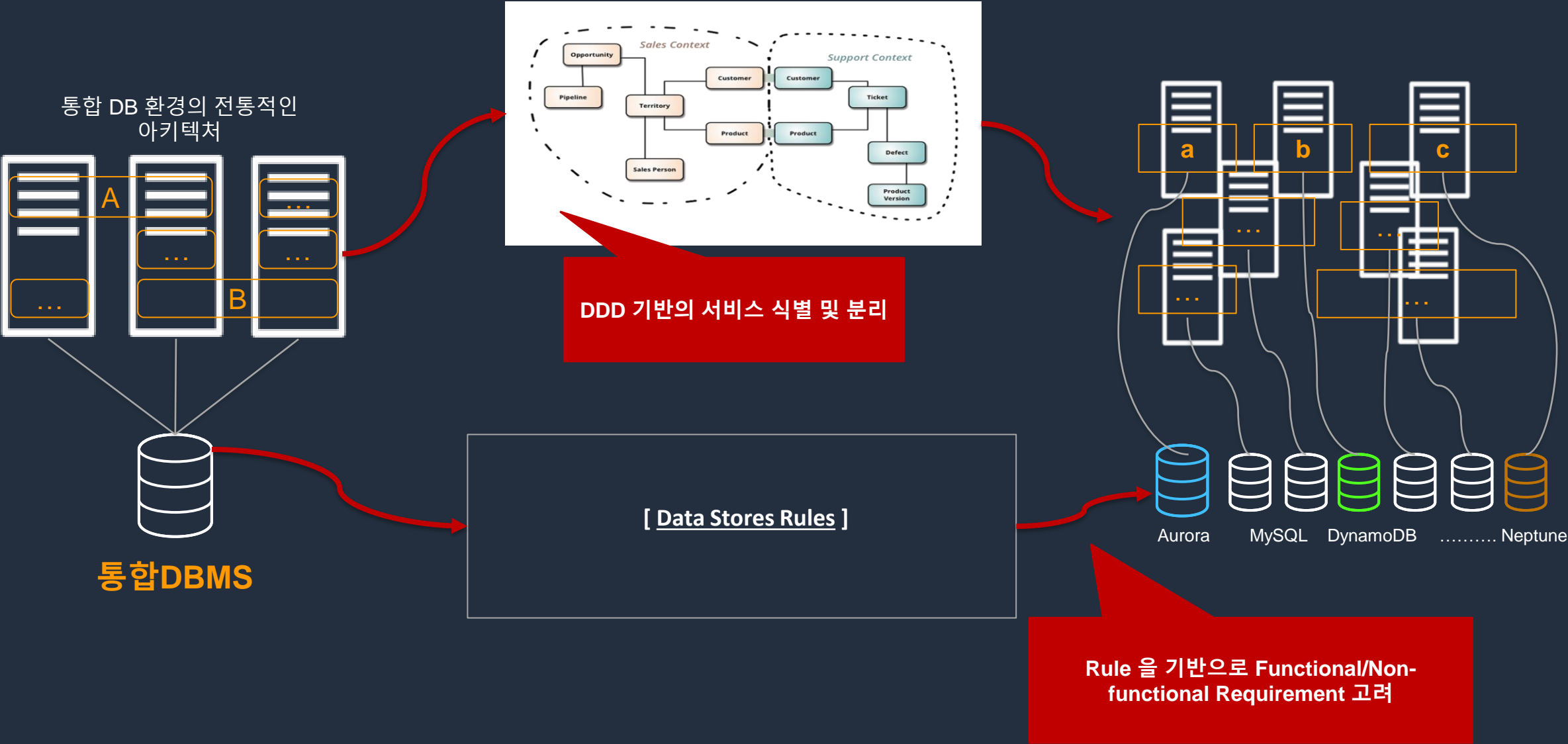
MSA Database Rules

1. 다양성 : RDBMS, NoSQL, Object Store...
2. 각 서비스는 자신만의 데이터 스토어 갖음
3. 각 서비스의 데이터 스토어는 서로 독립적
4. Denormalization 을 적극 활용
5. 조회 및 분석을 위한 별도의 통합 데이터 스토어를 고려



DBMS Modernization Guide

DDD 기반의 Data Store



DBMS Modernization Guide

DBMS Non-functional Requirements

- Special Type
- Size Limit
- Operator
- Integration
-

Check!

	Model Requirements	Non-functional Requirements	Proposal Data Store
A	Schema-less, Batch	Flexibility, Latency	DocumentDB/DynamoDB
B	Schema-less, Batch	Flexibility, Latency	DocumentDB/DynamoDB
C	Schema-less, Mainly Read	Flexibility, Latency	DocumentDB/DynamoDB
D	OLTP, 3rd Party Integration	Data integrity	Aurora
E	OLTP, Massive Query	Compatibility, Cost	Aurora or S3



DBMS Modernization Guide

Non-functional Requirements Matrix

Service	Latency	Durability	Storage	Availability	Recoverability from AZ Failure (RPO, RTO)
RDS	< 100 ms	99.99 (EBS)	64(16) TB (Depends on RDS Instance EBS Limits)	99.95	0s and 90s (Multi-AZ)
Aurora	< 100 ms	> 99.9	128 TB	99.99	0s and < 30s (Multi-AZ)
Aurora + ElastiCache	< 1 ms	> 99.9	128 TB	99.95	0s and < 30s (Multi-AZ)
DynamoDB	< 10 ms	> 99.9	No Limit	99.99	0s and 0s
DynamoDB / DAX	< 1 ms	> 99.9	No Limit	99.99	0s and 0s
ElastiCache Redis	< 1 ms	N/A	170.6 TiB	99.9	0s and < 30s (Multi-AZ)
Elasticsearch	< 200 ms	> 99.9	3 PB	99.9	0s and < 30s (Zone Aware)
S3	< 500 ms	99.999999999	No Limit	99.9	0s and 0s

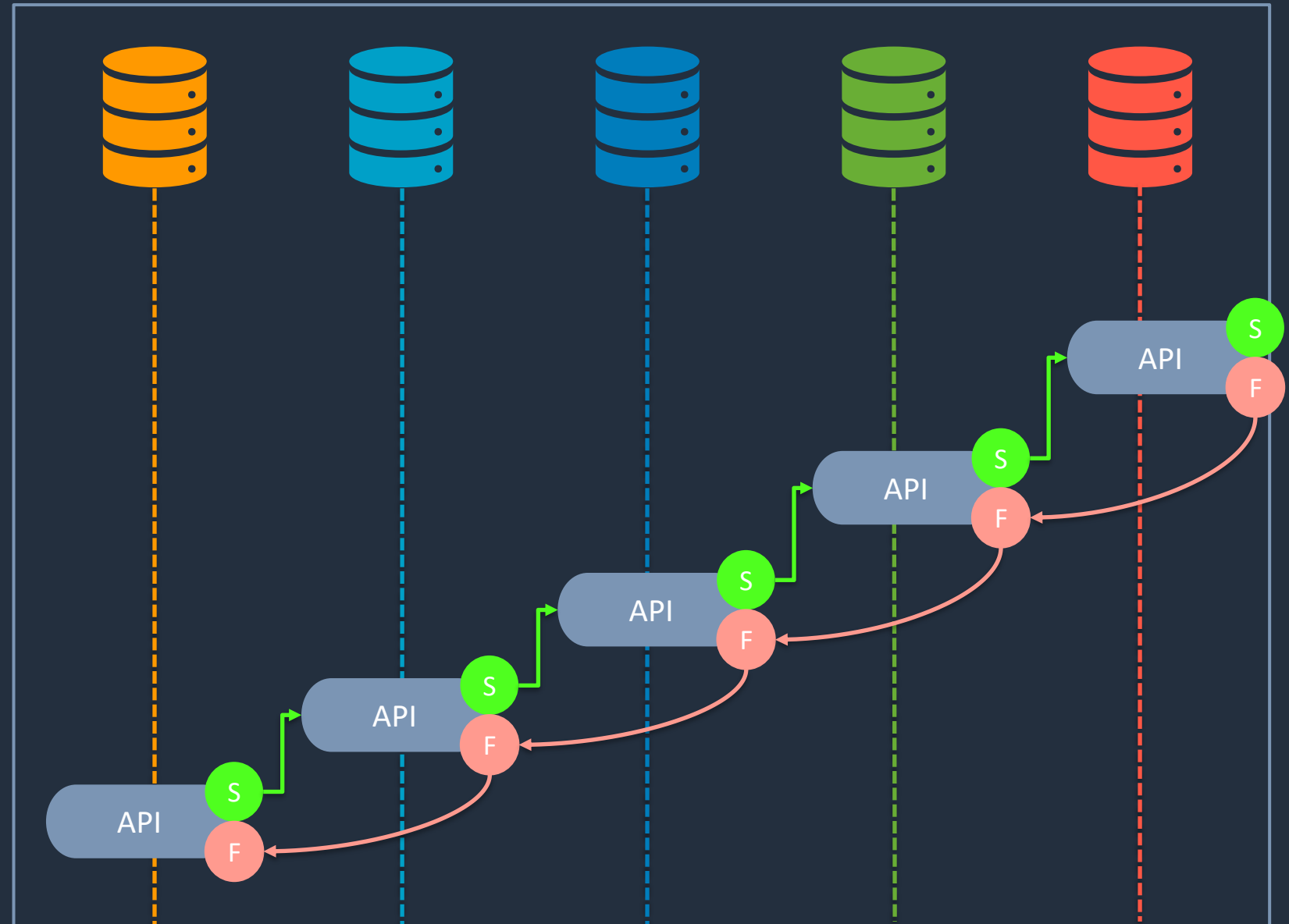


DBMS Modernization Guide

분산 Transaction 보장 방안 in MSA

SAGA 패턴을 통한 분산 트랜잭션 보장

- 각기 독립적인 서비스간의 분산 트랜잭션을 보장하기 위해 SAGA 패턴을 활용
 - 단계별 실행
 - 실패 시 이전 트랜잭션에 대한 보상 트랜잭션을 수행
 - 단계별 진행은 Message/Event를 통해 수행
 - Event Sourcing 기반하에서 활용



DBMS Modernization Guide

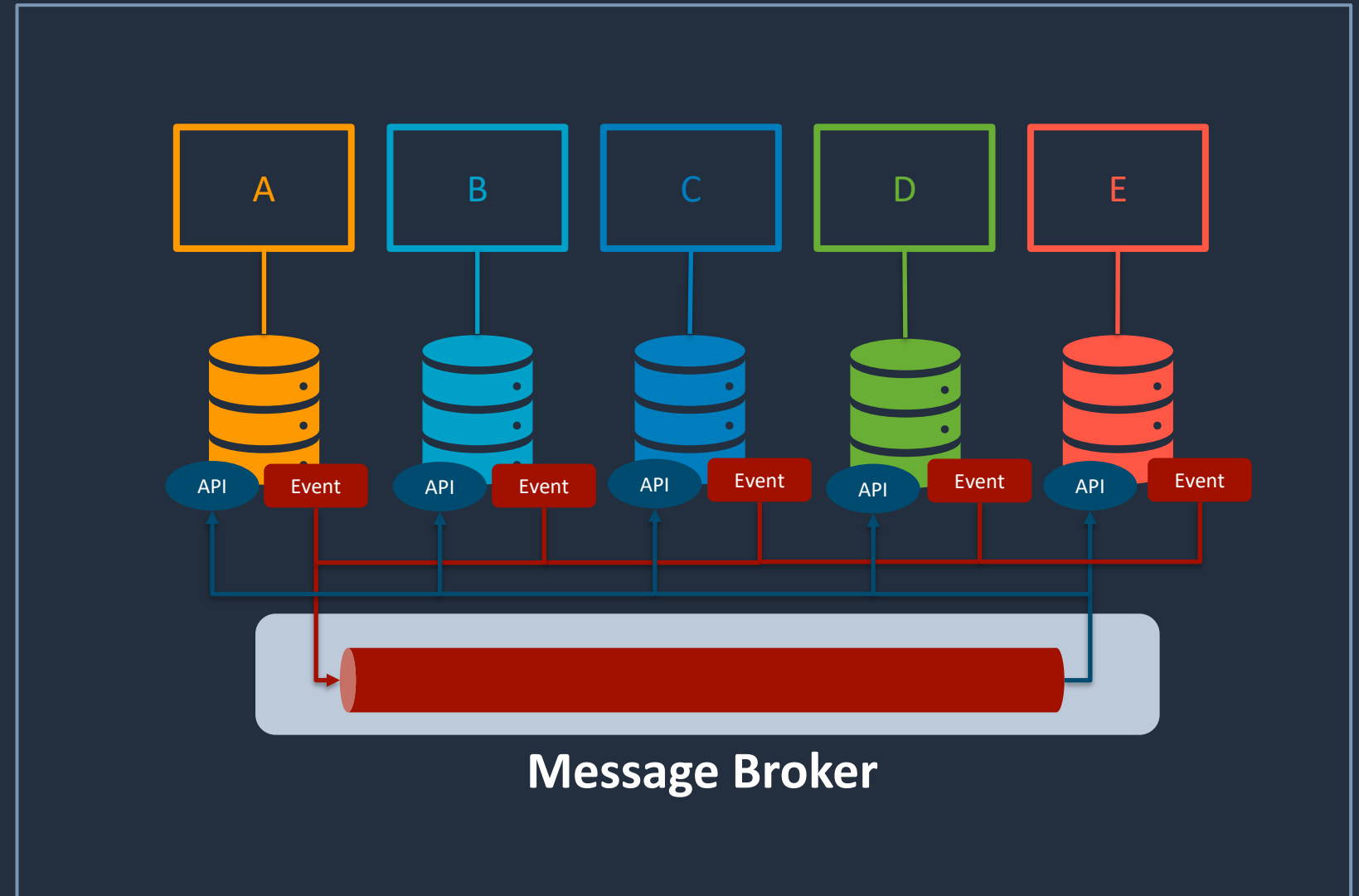
데이터 연계 방안

브로커 기반 메시징

- 느슨한 결합
- 메시지 버퍼링
- 유연한 통신
- 명시적 IPC

AWS 메시징 서비스의 장점

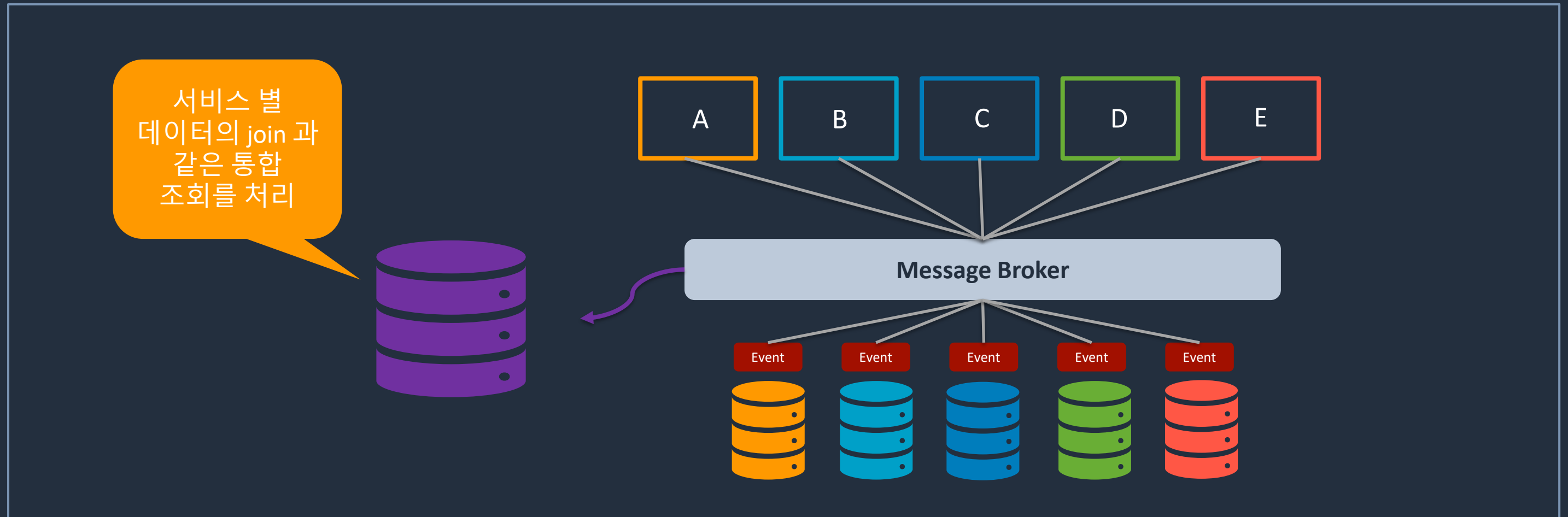
- 성능 병목 : 손쉬운 자원 증설
- 단일 장애 : 이중화 구성
- 운영 복잡도 : Managed Service 제공



DBMS Modernization Guide

용도에 따른 데이터 분리

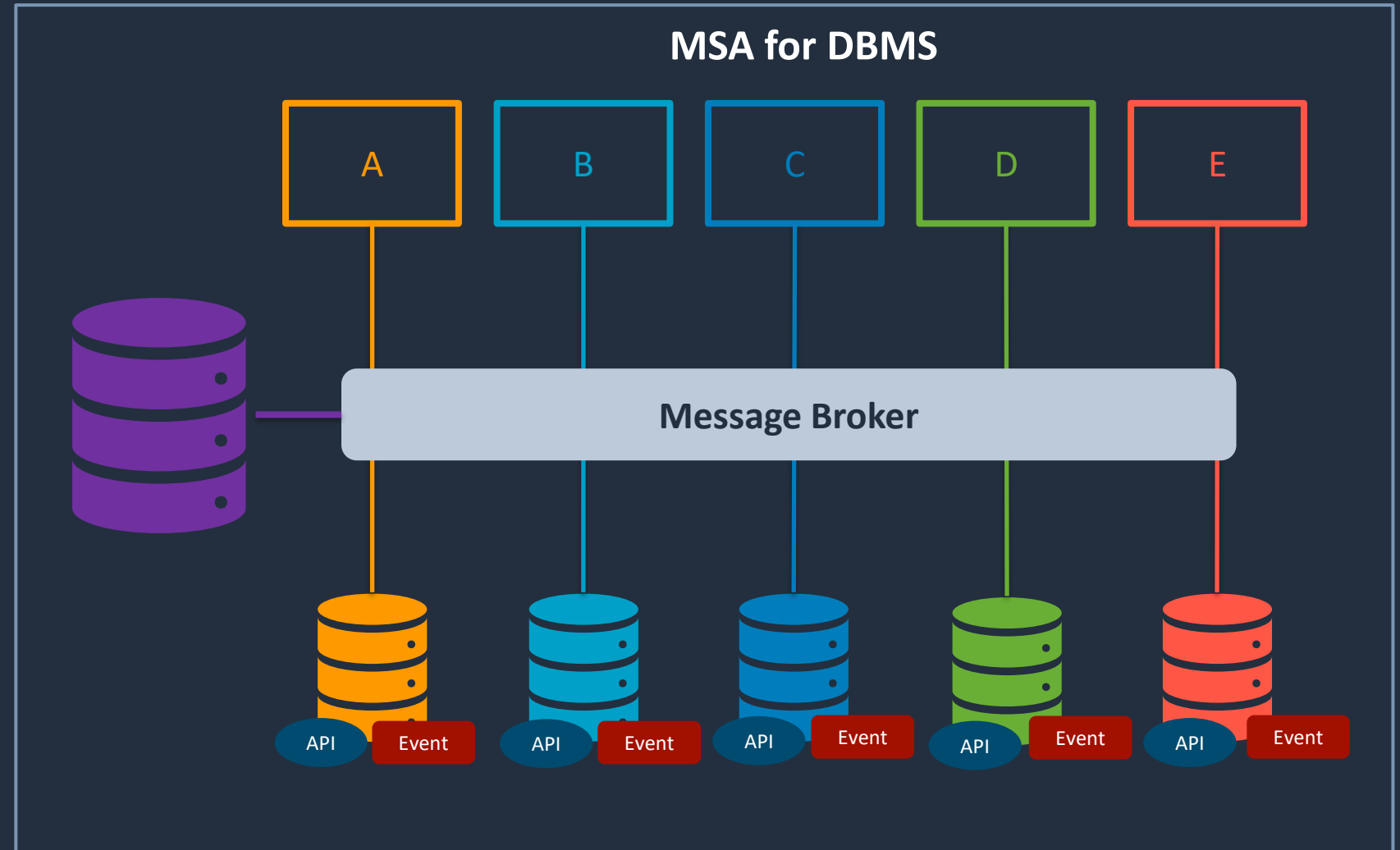
- 읽기와 쓰기를 분리하여 시스템 효율성을 극대화 -



DBMS Modernization Guide

기대 효과

- ✓ 시스템의 민첩성을 확보하여 비즈니스 변화에 빠르게 대응
- ✓ 각 서비스 별 의존성을 제거하여 유지보수의 편의성과 새로운 기능의 손쉬운 적용을 달성
- ✓ 상용 소프트웨어 라이선스 비용 절감 : DB Freedom
- ✓ 자동화된 관리/운영을 통한 효율성을 극대화



감사합니다.