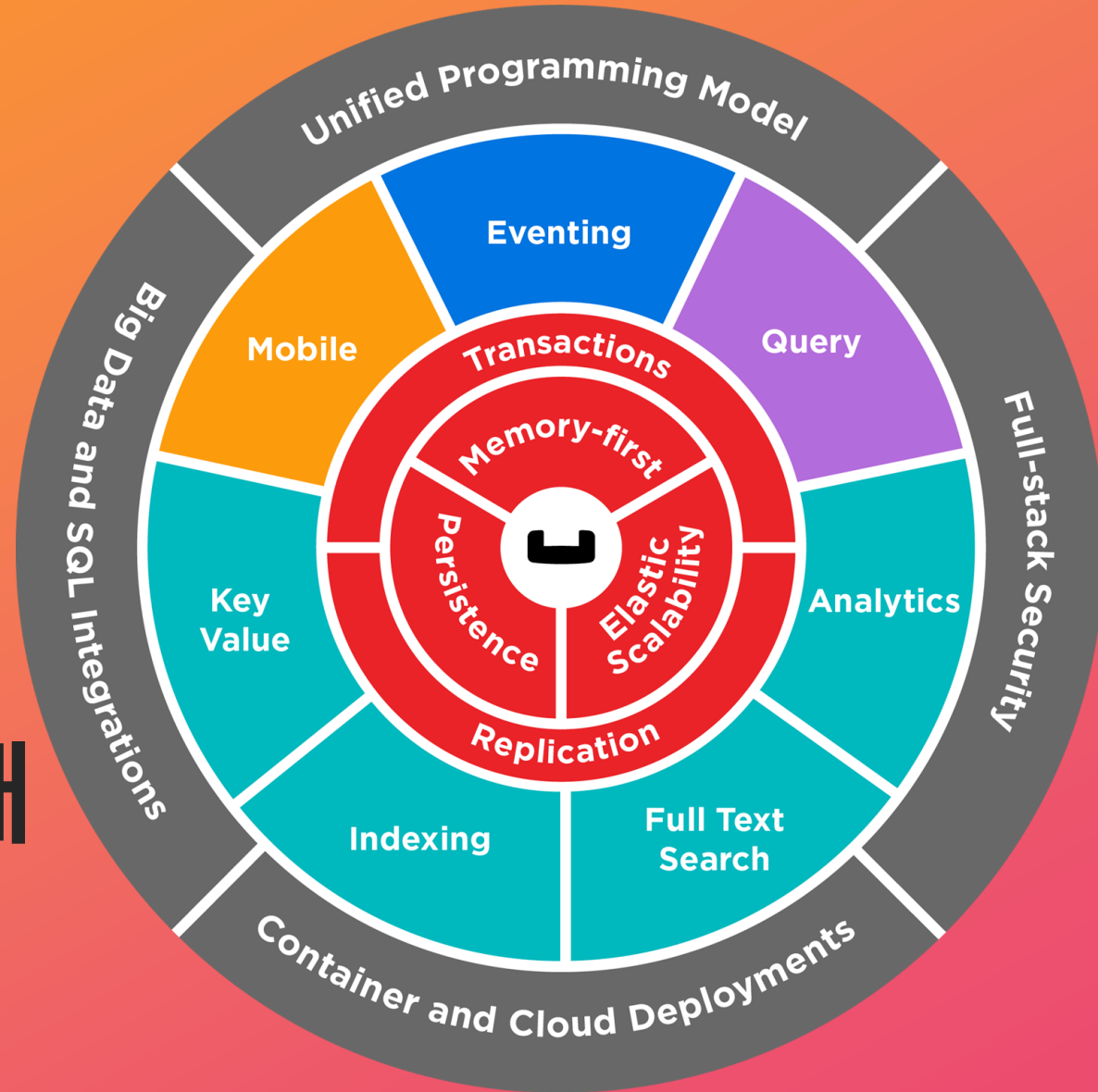


# 1 카우치베이스 소개





컨슈머(고객의 고객)는 단순 소비를 넘어,  
최고의 디지털 경험을 원함.



**Personalized**



**Feature-rich**



**Fast**



**Anywhere**



**Anytime**

# 이를 위한 모던 어플리케이션의 요구 사항

## 최상의 '경험'을 제공할 수 있어야

- 개인화된 & 즉각 반응하는
- 언제 어디서든
- 실시간 정보 & 재고
- 신뢰할만한 트랜잭션 처리

## 효율적인 적용이 가능해야

- 하이브리드&멀티 클라우드 지원, 엣지, 5G
- 100% 가용성 & 글로벌 스케일
- 유연한 관리 옵션
- 비용효율적인

## 개발이 용이해야

- 개발 단순화 및 가속화
- 기존 스킬셋 활용 (SDKs & SQL)
- 베스트 프랙티스 지원
- 데이터 무분별 확산 방지



# 데이터베이스 요구사항의 변화

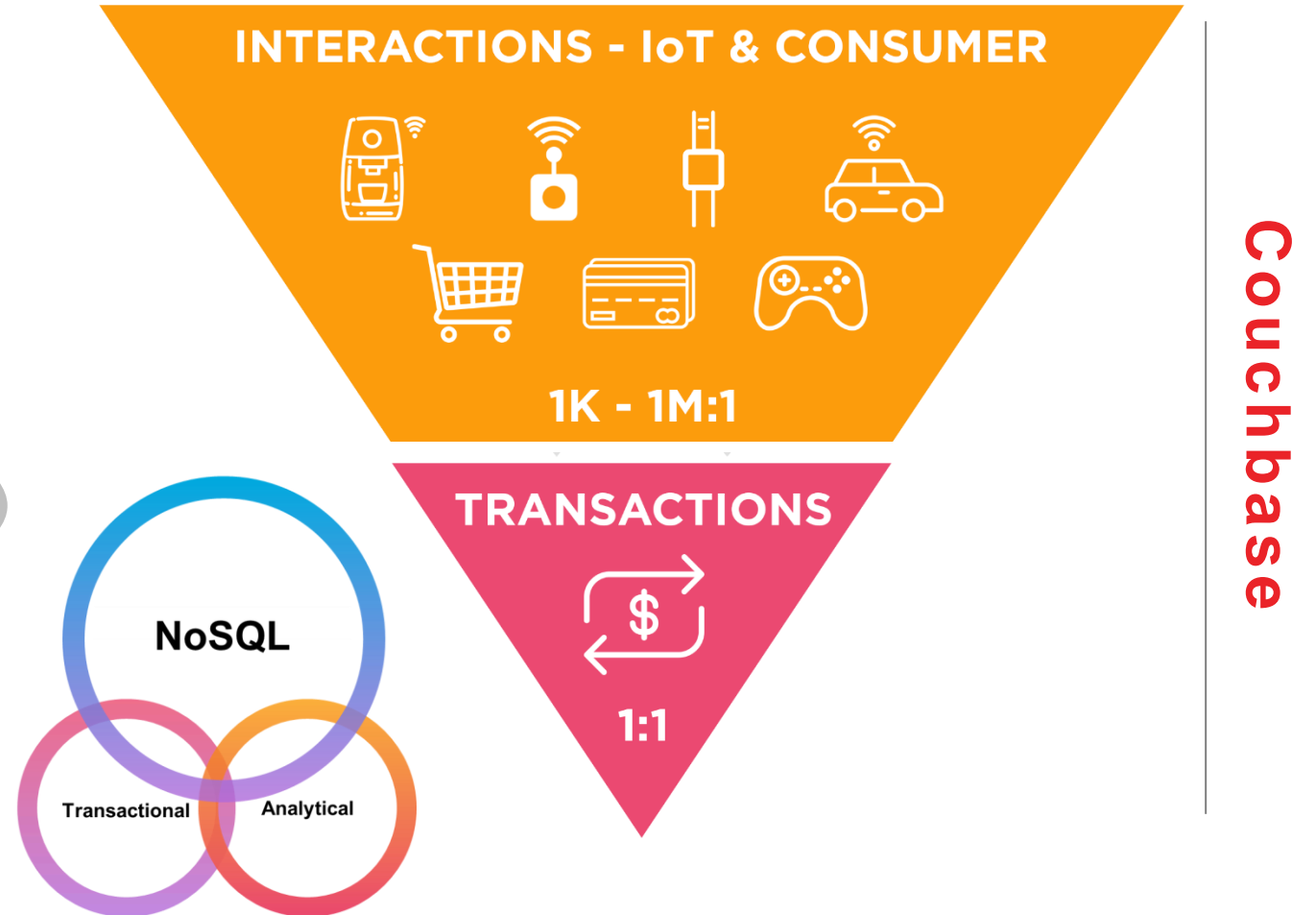


## Modern Demands (NoSQL)

- 대규모 데이터와 사용자
- 확장에 따른 성능 향상
- 민첩한 개발을 위한 유연성
- 마이크로서비스 아키텍처 (MSA)
- 웹, 모바일 & IoT 경험

## Legacy Demands (Relational)

- 소수 사용자
- 주로 단일 서버내에서 운영
- 트랜잭션, 일관성, ACID보장, 정합성
- 스토리지 효율성 중요

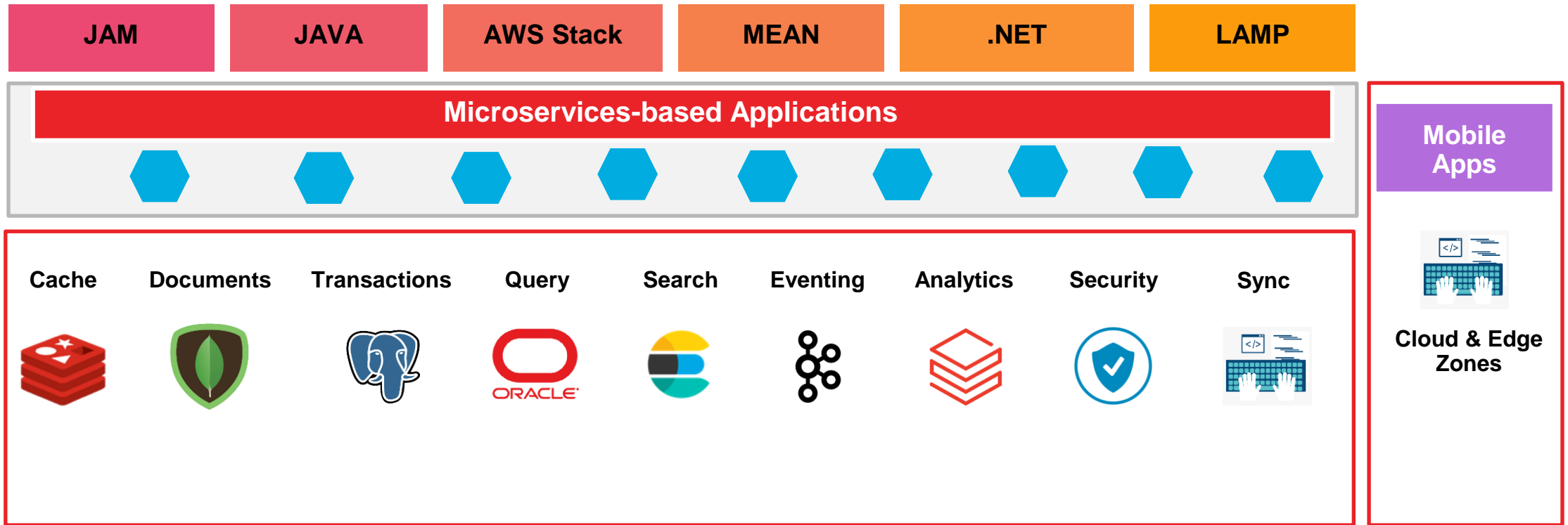


“The amount of data created over the next 3 years will be more than the data created over the past 30 years.” May 2020



# 2000년대의 일반적인 설계 방향

Data sprawl (무분별한 데이터 확산)을 수용.



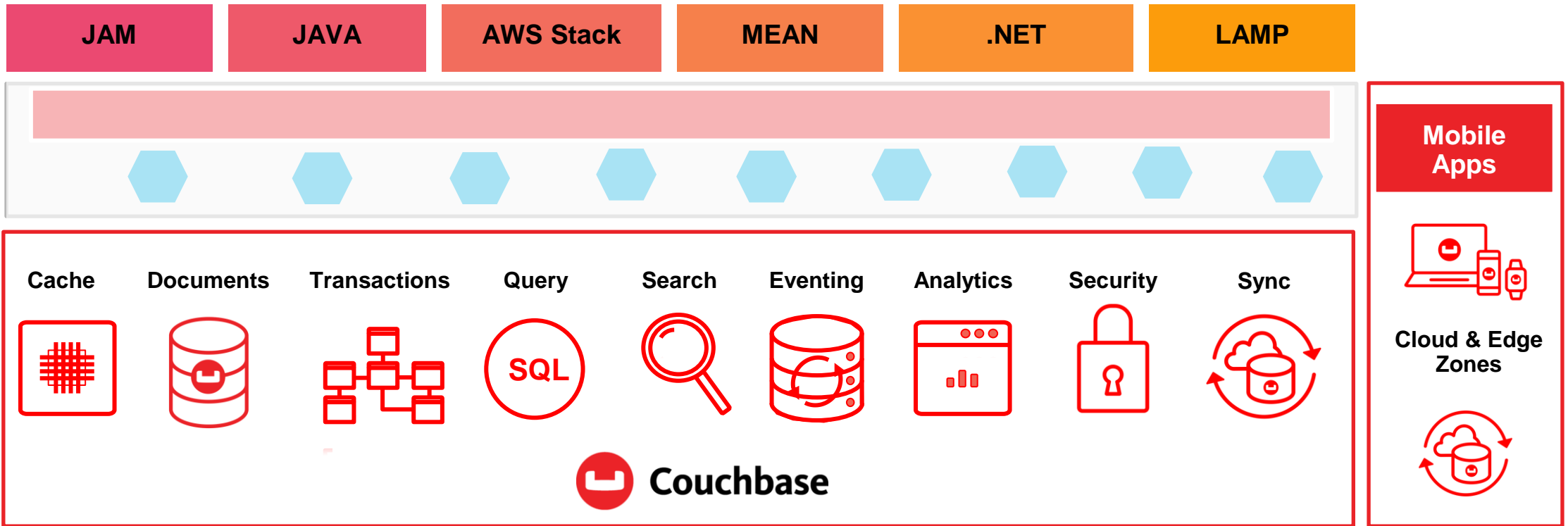
데이터 스프롤은 개발자가 어플리케이션에 필요한 여러 기능을 사용하기 위해, 다수의 데이터베이스를 선택하면서 발생함.

이로 인한 복잡성, 비용증가, 관리노력으로 결국 릴리스 사이클이 늦어지는 결과로 나타남.

# 멀티모델을 지원하는 새로운 데이터베이스의 출현 – 카우치베이스



데이터 관련 모든 기능은 지원 하지만 단일 플랫폼으로 제공.



중복 감소






용이한 관리

빨라진 릴리스 사이클

낮아진 운영 및 인프라 비용

# 통합 데이터베이스 플랫폼



| 유연한(Flexible)  | 익숙한(Familiar)   | 고속의(Fast)  | 가성비좋은 Affordable)   | 미래기술수용 (Futureproof)   |
|--|---|--|---|--|
|  <ul style="list-style-type: none"> <li>• JSON 문서</li> <li>• 멀티 모델 서비스</li> <li>• 어느곳에서나 구현가능</li> <li>• 모바일 &amp; 엣지까지 지원</li> </ul> |  <ul style="list-style-type: none"> <li>• SQL++ 쿼리 언어</li> <li>• 다이내믹 스키마</li> <li>• ACID SQL 트랜잭션</li> <li>• 코스트 기반 옵티마이저</li> <li>• 12개 주요 언어지원 (모바일포함)</li> </ul> |  <ul style="list-style-type: none"> <li>• 메모리 퍼스트 설계</li> <li>• 클라우드 네이티브 확장</li> <li>• XDCR을 통한 지역간 복제</li> <li>• HA, DR &amp; 백업</li> <li>• Cloud – Edge간 저지연 (Low latency) 지원</li> </ul> |  <ul style="list-style-type: none"> <li>• 일래스틱 클러스터 확장, 자동 샤딩 &amp; 자동재분배</li> <li>• 분리가능한 멀티 디멘션 확장지원</li> <li>• 고집적 스토리지 지원</li> <li>• 업계 최고수준의 가성비</li> </ul> |  <ul style="list-style-type: none"> <li>• Fully-managed DBaaS지원 (AWS/GCP/Azure)</li> <li>• Self-managed 쿠버네티스 자동 오퍼레이터 지원</li> <li>• Self-managed 모바일 &amp; 엣지</li> </ul> |
| <p><b>카우치베이스 관리형 &amp; 고객 자체 관리형 클라우드 지원</b></p>   |   |  |   |  |



# 전통적 RDB의 강점과 현대적인 NoSQL의 강점을 견비

관계형 vs NoSQL 데이터베이스 논쟁은 더 이상 의미없음.

## 관계형 DB 강점 수용 & 강화

- 유연한 데이터 스키마 (Schema-less도 지원)
- 엔터프라이즈급 보안
- 분산 SQL 트랜잭션 처리
- SQL++ 쿼리 언어
  - 조인지원(Collections & Documents 활용)
  - UDF지원 (PL/SQL용)
  - 코스트 기반 쿼리 옵티마이저
  - 쿼리 메모리 쿼터 할당
- CDC/이벤팅/스트리밍지원
- 인덱스 어드바이저 (in-database & online)
- 광범위한 SDK 지원
- 태블로(BI)통합

## NoSQL의 유연함과 성능

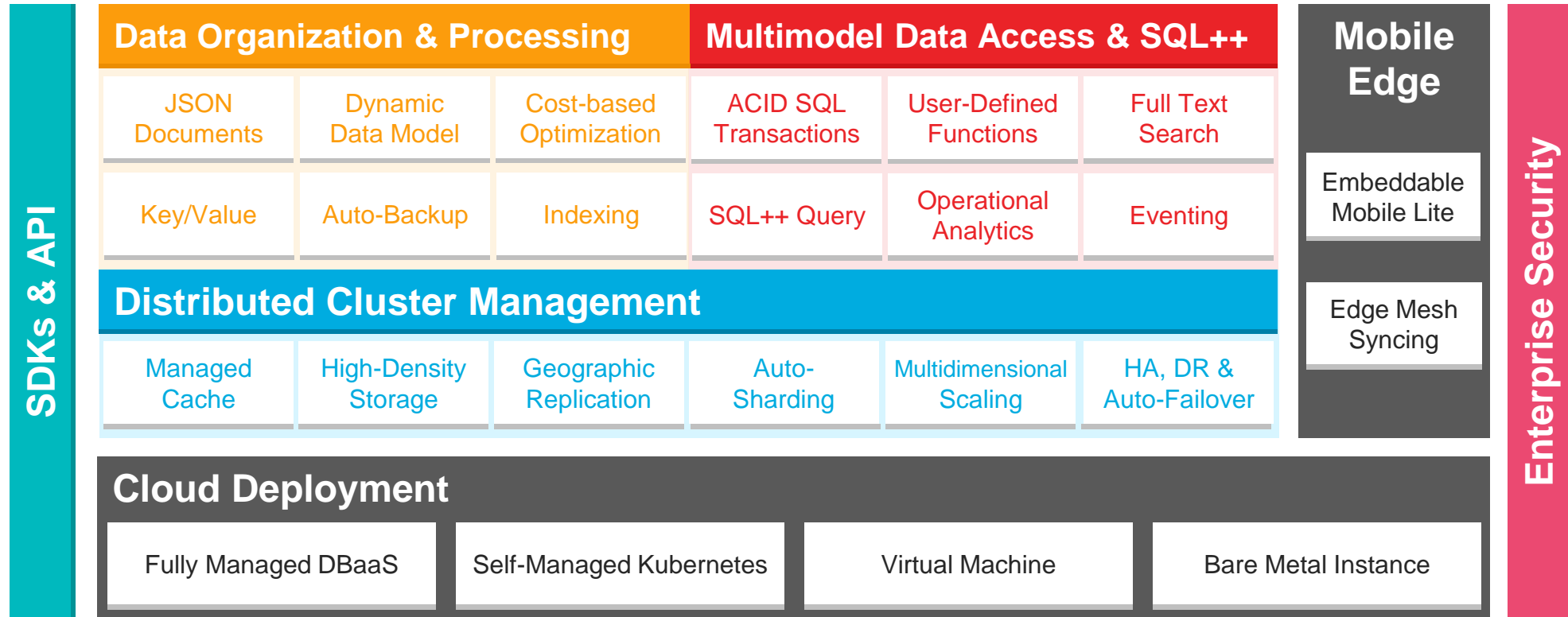
- 멀티모델 액세스
- 멀티 디멘전 스케일링 (MDS)
- 메모리 퍼스트 아키텍처
- 100% 업타임 보장
- 모바일 & 엣지 & 실시간 연계
- 어느곳에서나 설치가능-클라우드, 엣지, 임베딩
  - Capella DBaaS (벤더관리형 DB PaaS)
  - Shared-nothing, scale out architecture
  - 자동 파티셔닝, 자동 데이터 재분배
  - 백업지원
- 클라우드 네이티브 쿠버네티스 기반 지원 및 자동화
- Kafka/Spark/Elasticsearch 지원 Connector



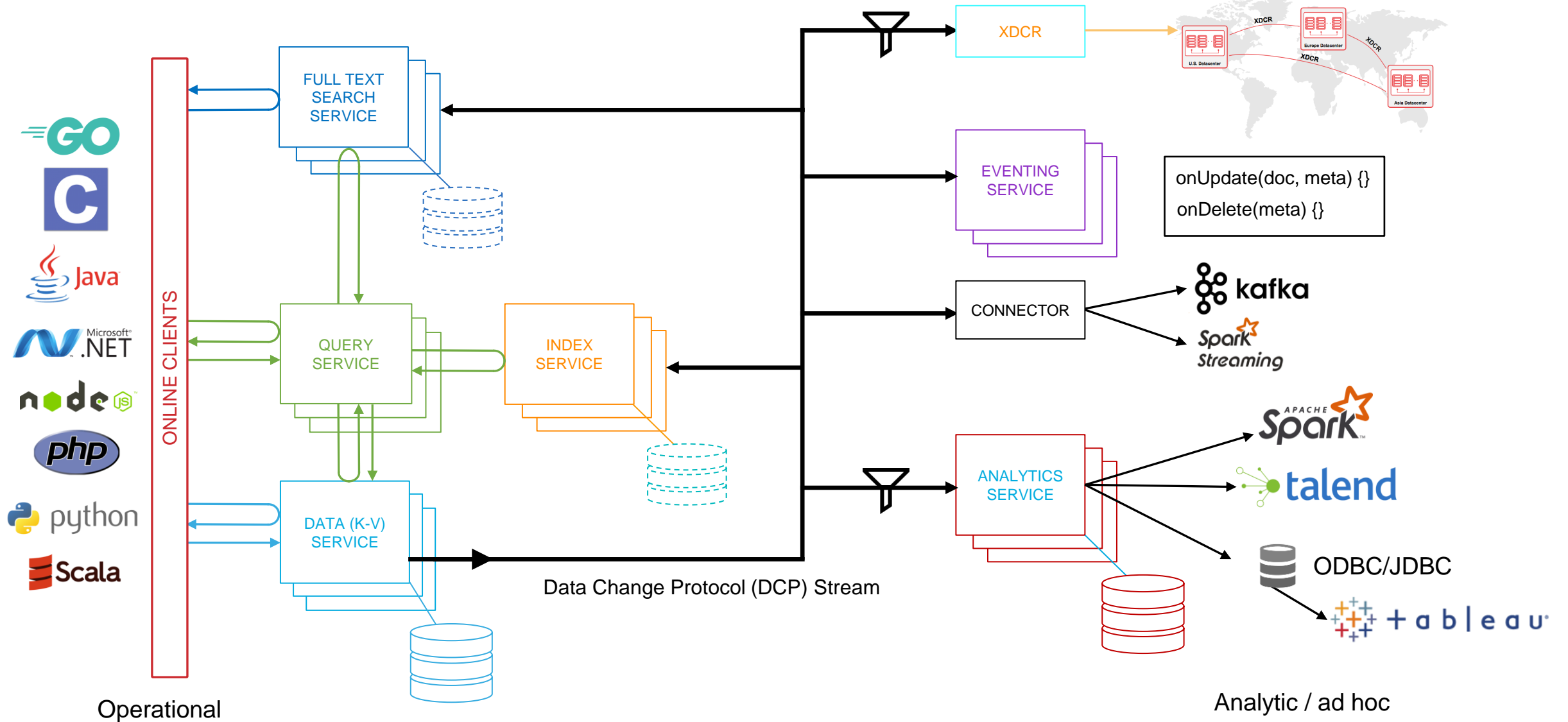


# 플랫폼 아키텍처: 유연함, 익숙함, 빠른 성능과 접근성

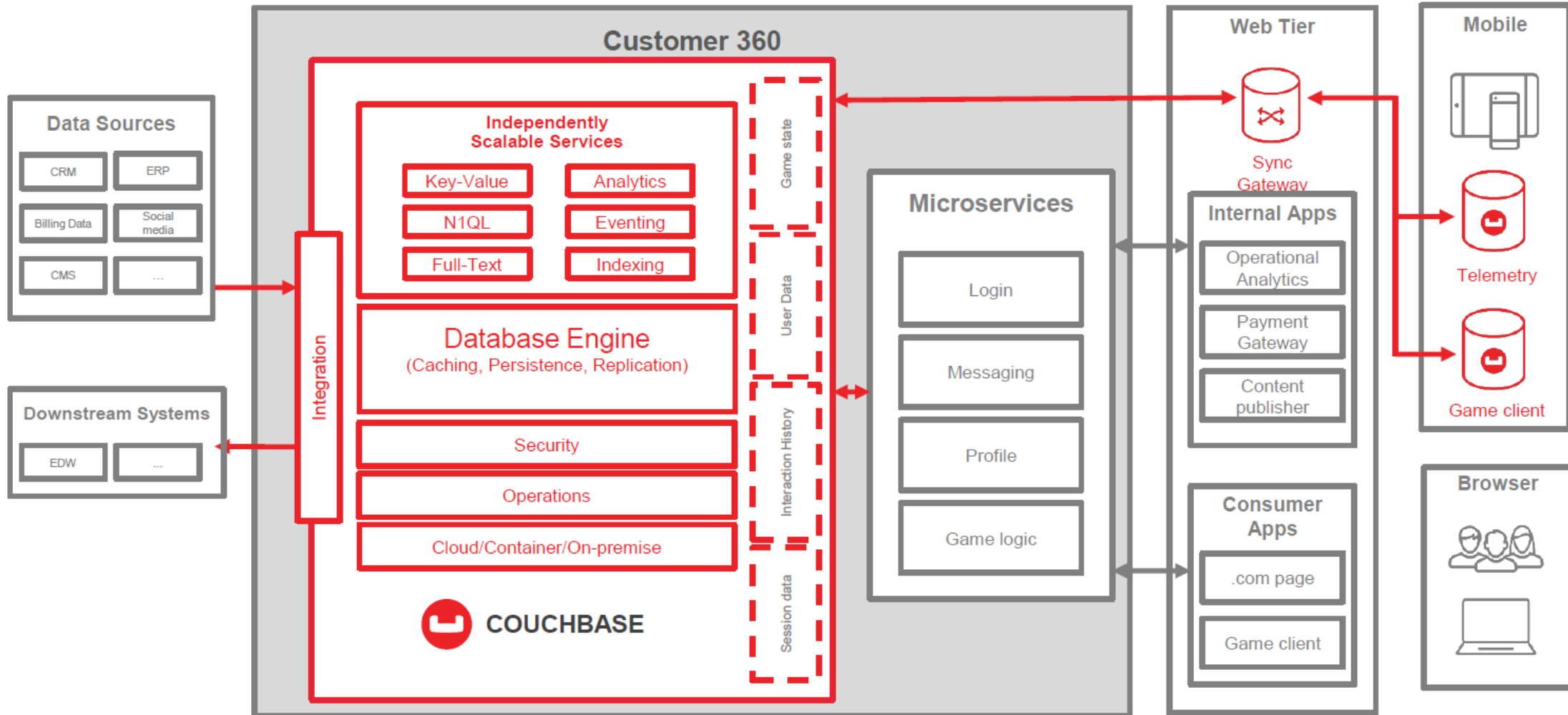
미래기술수용: JSON의 유연함 + RDBMS의 익숙함 + 인메모리의 고성능 + 클라우드의 효율성



# 주요 서비스 아키텍처(트랜잭션, 쿼리, 검색, 분석, 이벤트 등)



# 레퍼런스 아키텍처 (예, Customer 360)



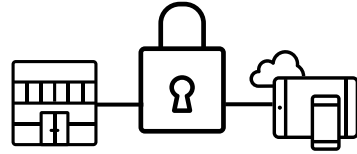


# 신뢰할 수 있고, 안전한 그리고 확장가능한 구성



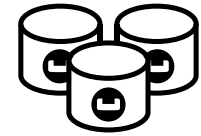
- 업무 부하에 따라 각각의 서비스별로 자원 확장 가능 (Scale up & Scale out가능)
- 가장 실제적인 & 예측가능한 자원/예산정책수립 (MSA 대응)

## 멀티 디멘전 확장 (MDS)



- Full-stack 보안
- 모바일 스토리지/네트워크상/브라우저내/클러스터/영구저장디스크 장치에 이르는 전 과정에서 보안 지원

## 엔터프라이즈 보안



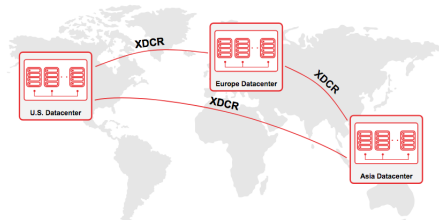
- 자동 파티셔닝 & 데이터 재분배
- Shared-nothing 일래스틱 아키텍처
- 고가용, 내장애, 분산 클러스터 지원.

## 액티브/액티브 클러스터링



- 각각의 SDK에 탑재된 트랜잭션 관리
- 분산 클러스터상의 '커밋'
- OLTP/원장관리 사례 지원

## 분산 ACID 트랜잭션 지원



- 각 사용자들에게 저지연 액세스 제공
- 로컬 규제를 위한 geo-filtering 구현
- 마이그레이션 & 백업 & hybrid/on-prem/cloud 구축에 활용

## 지역간 복제 (XDCR)



- 쿠버네티스 기반의 command & control
- 자동 배치, 확장, 노드 업그레이드지원
- 최고 수준의 클러스터 관리 & 리포팅
- AKS, GKS, Azure, Red Hat Openshift

## 자동 오퍼레이터 오케스트레이션

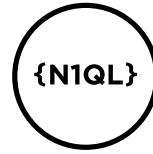


# 이를 통한 멀티 모드, 독립적인 서비스를 지원



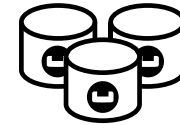
- JSON포맷을 통한 Keys, Values, Arrays 변경의 용이성.
- Normalized, denormalized & binary data
- 카우치베이스 전체 서비스의 쿼리지원

## JSON 스키마 유연성



- JSON용 SQL
- 완전한 CRUD 지원
- 익숙한 syntax;  
SELECT, WHERE, JOIN, INSERT, DELETE

## N1QL로 익숙한 SQL지원



- 쿼리 서비스: 각종 요청 중재
- 인덱싱 서비스: 캐쉬 우선, 디스크 나중
- 데이터 컨트롤: write/retrieve data

## 메모리 퍼스트 데이터 처리 서비스



- 역 인덱스 (Inverted Indexes)
- Language 인지
- 스코어링

## 텍스트 검색



- ETL없이 실시간 데이터 복제, MPP 쿼리 실행
- 대용량 데이터에 대한 복합쿼리수행
- 비즈니스 팀을 위한 리얼타임 분석력 제공

## 분석 서비스

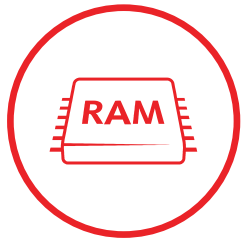


- 자바스크립트 기반의 캡처 & 공지
- Kafka 연동 스트리밍지원
- 고유의 CDC를 적용가능

## 이벤팅 서비스



# 카우치베이스 주요 차별화 요소



메모리-퍼스트  
아키텍처



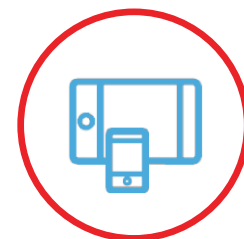
SQL/SQL++  
& 텍스트 검색지원



워크로드 분리  
(MDS)



데이터 센터간  
복제



모바일 &  
엣지

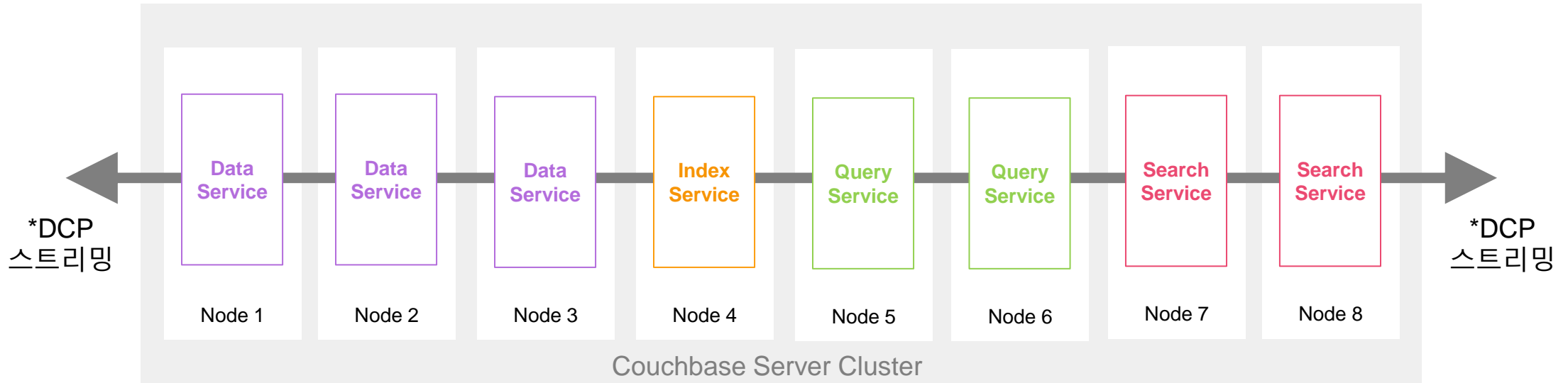


클라우드  
네이티브



# 메모리 퍼스트 아키텍처

디스크 병목 발생이 없는 데이터 이동



- 모든 구성 요소간 인-메모리 스트리밍을 통한 업데이트
- 인메모리 캐시
- 메모리전용 데이터 버킷(bucket)
- 메모리전용 인덱스

\*Data Change Protocol : 카우치베이스 자체 용어, 데이터 변경 내부 공유 프로토콜



# SQL++ 지원 : 조인/UDF지원/ACID보장(멀티 다큐먼트포함)

익숙한 SQL 지원으로 개발자/기술인력등의 스킬셋 활용가능

• 잘 알려진 쿼리 syntax ;

- ANSI standards
- JOINS
- Sub-queries
- Nested objects
- Arrays

```

Relational SQL

SELECT ac.industry,
SUM(CASE WHEN a.activitytype = 'Task'
THEN 1 ELSE 0 END ) task,
SUM(CASE WHEN a.activitytype
='Appointment'
THEN 1 ELSE 0 END ) appts
FROM crm.activity a
INNER JOIN crm.account ac
ON (a.accid = ac.id)
WHERE a.startdate BETWEEN '2018-10-01'
AND '2018-12-31'
GROUP BY ac.industry

```

```

Couchbase - SQL++

SELECT ac.industry,
SUM(CASE WHEN a.activityType = 'Task'
THEN 1 ELSE 0 END) task,
SUM(CASE WHEN a.activityType =
'Appointment'
THEN 1 ELSE 0 END ) appts
FROM crm a
INNER JOIN crm ac ON a.accid = ac.id
AND ac.type='account'
WHERE a.type='activity'
AND a.startDate BETWEEN '2018-10-01'
AND '2018-12-31'
GROUP BY ac.industry

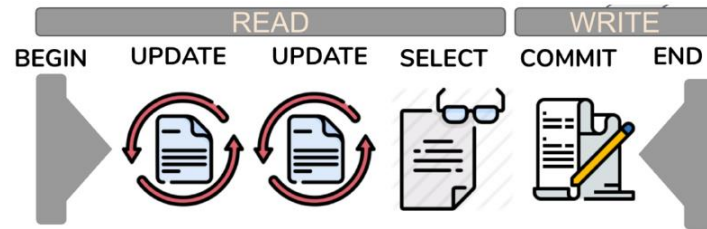
```

```

MongoDB Query Language

db.activity.aggregate(
{ $match: { startDate: { $gt: '2018-01-01',
$lt: '2018-12-31' } } },
{ $lookup: {
from: "account",
localField: "accid",
foreignField: "id",
as: "account_docs"
}
},
{ $match: { "account_docs": { $ne: [] } } },
{ $unwind: "$account_docs" },
{
$project: {
item: 1,
task: { $cond: { if:
{ $eq: ["$activityType", "Task"] }, then: 1,
else: 0 } },
appt: { $cond: { if:
{ $eq: ["$activityType", "Appointment"] },
then: 1,
else: 0 } } }
},
{
$group: {
_id: "$account_docs.industry",
tasks: { $sum: "$task" },
appointments: { $sum: "$appt" }
}}
);

```



```

BEGIN WORK; -- {stmt:s1}
UPDATE x1 SET a = a + 1 WHERE b < 10; -- {stmt:s2}
UPDATE x1 SET a = a + 15 WHERE b < 10; -- {stmt:s3}
SELECT a, b, c FROM x1 WHERE b < 20; -- {stmt:s4}
COMMIT WORK; -- {stmt:s5}

```

[ACID지원]

| Feature    | SQL | N1QL |
|------------|-----|------|
| JOIN       | ✓   | ✓    |
| TRANSFORM  | ✓   | ✓    |
| FILTER     | ✓   | ✓    |
| AGGREGATE  | ✓   | ✓    |
| SORT       | ✓   | ✓    |
| SUBQUERIES | ✓   | ✓    |
| PAGINATION | ✓   | ✓    |
| OPERATORS  | ✓   | ✓    |
| FUNCTIONS  | ✓   | ✓    |

[SQL유사도 비교]





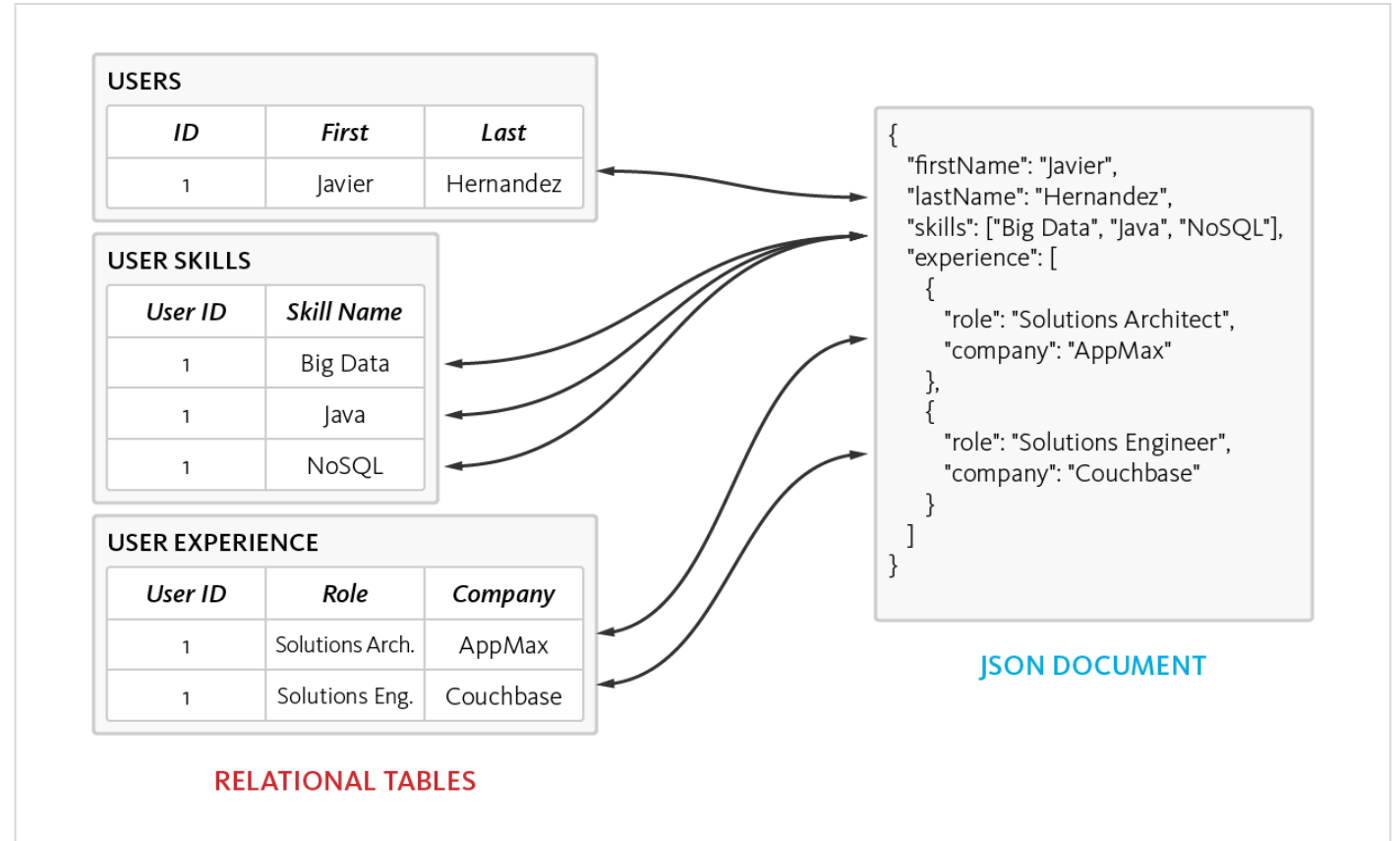
# JSON 스키마의 유연성

## 다양한 방식으로 데이터 저장:

- Denormalized single document
- Normalized with references
- Add new values when needed
- Support for binary values

## 다양한 방식으로 데이터 액세스:

- Direct Key-Value
- SQL querying
- Full-Text Search
- MPP for large, ad-hoc access
- Event-based triggers

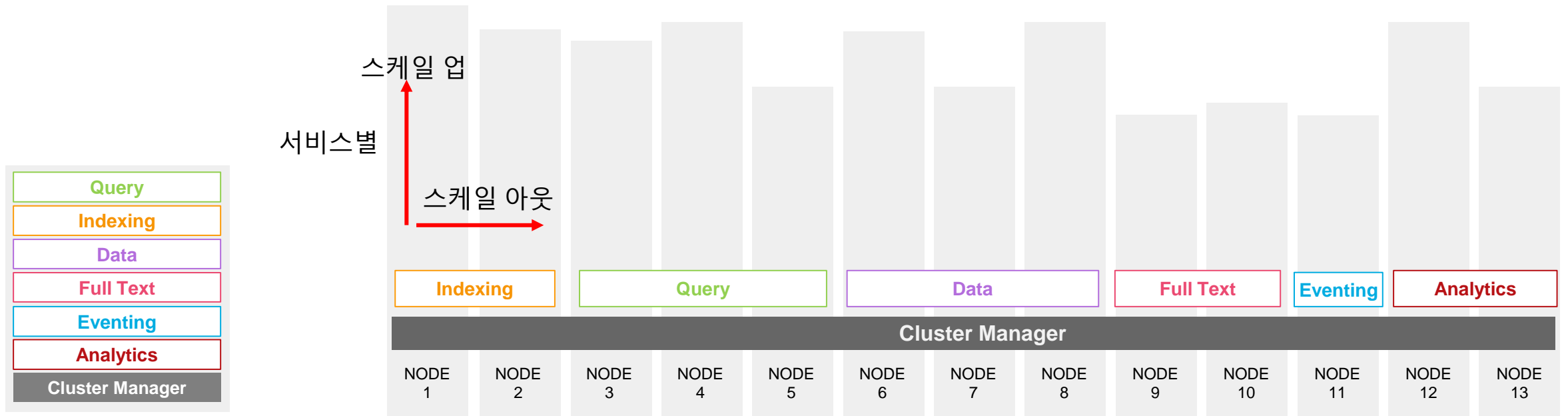


# 탄력적 스케일링: 각각 분리된 워크로드 지원 (자원사용과 확장 최적화)

각각의 서비스를 부하에 따라 독립적으로 확장하여, 자원운용성 및 비용 예측성을 최적화함.

## 개발 & 테스트 환경

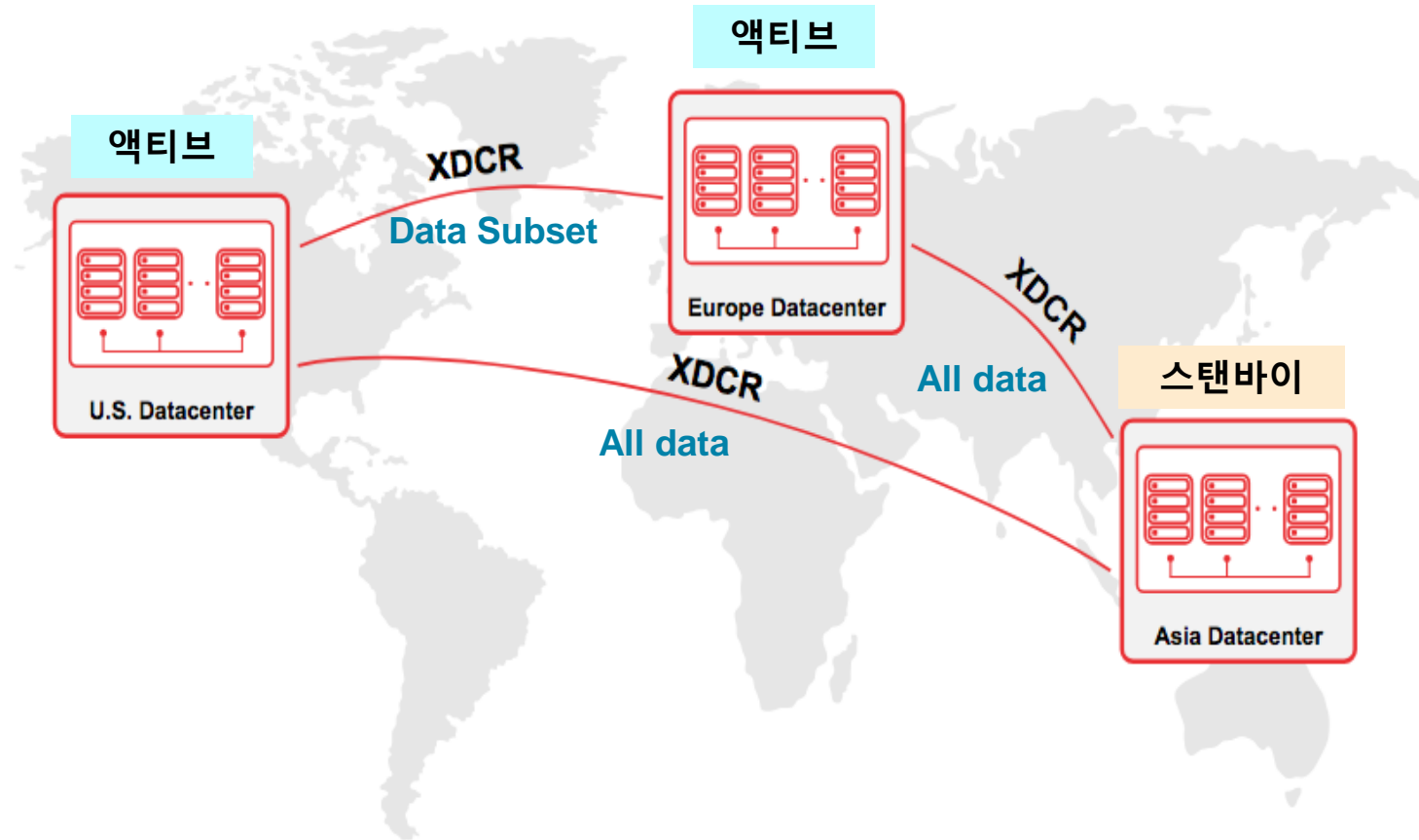
## 운영 환경





# 고가용성을 위한 데이터 센터간 복제지원 (XDCR)

XDCR을 통한 지역간 복제등이 가능하며 또한 데이터 발생지 귀속에 대한, GDPR 요건도 수용가능함.





# Couchbase Lite: 모바일/엣지/임베딩 용도의 단일 NoSQL DB

## Developer Friendly

Any Platform  
Any Language  
Any Device



EDGE DEVICES



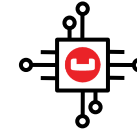
Smartphone



Tablet



Laptop



IoT Device



Smartphone



Tablet

*Devices running Couchbase Lite (Peer-to-Peer Sync)*



iOS and macOS



Kotlin for Android



Java for Android



Java Desktop



Embedded, Desktop & Mobile



Microsoft .NET Desktop & Tablets



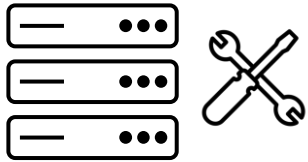
Xamarin iOS & Android

- **C-API**를 통해 카우치베이스 라이트를 어떤 임베디드 어플리케이션에도 탑재가능
- **SQL++ (SQL for JSON)**와 **API지원로 개발 생산성 증대**
- **Schema-less JSON 포맷**을 통한 개발 및 어플리케이션 업그레이드 단순화

# 클라우드 네이티브 + 하이브리드 지원



## PRIVATE CLOUD



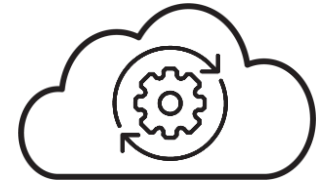
- 고객에 의한 관리 및 운영
- 최고 수준의 유연성, 퍼포먼스, 커스터마이제이션 제공
- DBA와 인프라운영인력에 의한 관리
- 쿠버네티스 구성 지원

## PUBLIC CLOUD



- 고객에 의한 관리 및 운영
- CSP사의 템플릿과 Terraform을 통해, 퍼블릭 클라우드내 구성
- 커스터마이제이션과 고성능제공
- DBA와 인프라운영인력에 의한 관리
- 쿠버네티스 구성 지원

## -FULLY MANAGED AS-A-SERVICE



- 카우치베이스 클라우드
- 자동 구성, 스케일링, 장애복구 및 업그레이드
- 통합된 모니터링과 얼러팅
- 멀티 클라우드 및 하이브리드 구성 지원
- 엔드 투 엔드 보안: '전송중' 데이터, '저장된' 데이터 보안 적용

유연성, 커스터마이제이션  
및 고성능



단순성, 고성능 및  
사용 편의성



# 통합된 프로그래밍 인터페이스 지원

각각의 카우치베이스 서비스에 대한 통일된 API를 제공하는 카우치베이스 SDK를 어플리케이션에 임베딩.

COUCHBASE  
CLIENTS



## Key Value

```
INSERT("key1",doc)
GET("key1")
```

## N1QL query

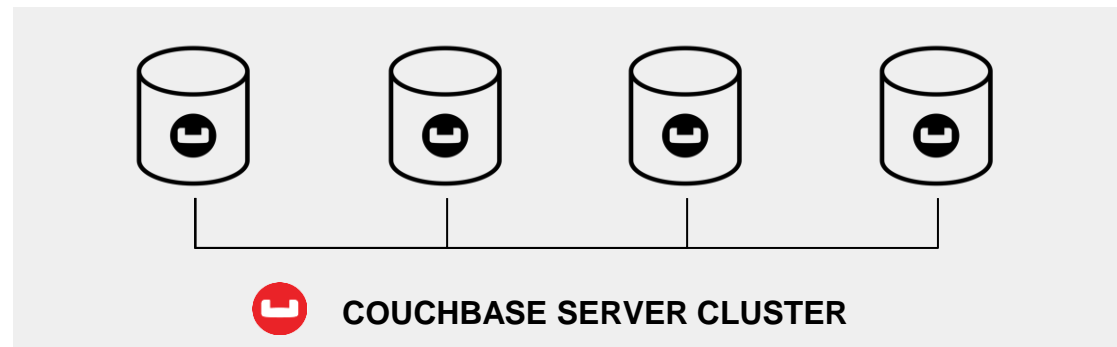
```
SELECT a.country
FROM airline a
WHERE a.name = "Excel
Airways";
```

## Full Text Search

```
{
  "fields": ["*"],
  "query": {
    "query": "+nice +view"
  }
}
```

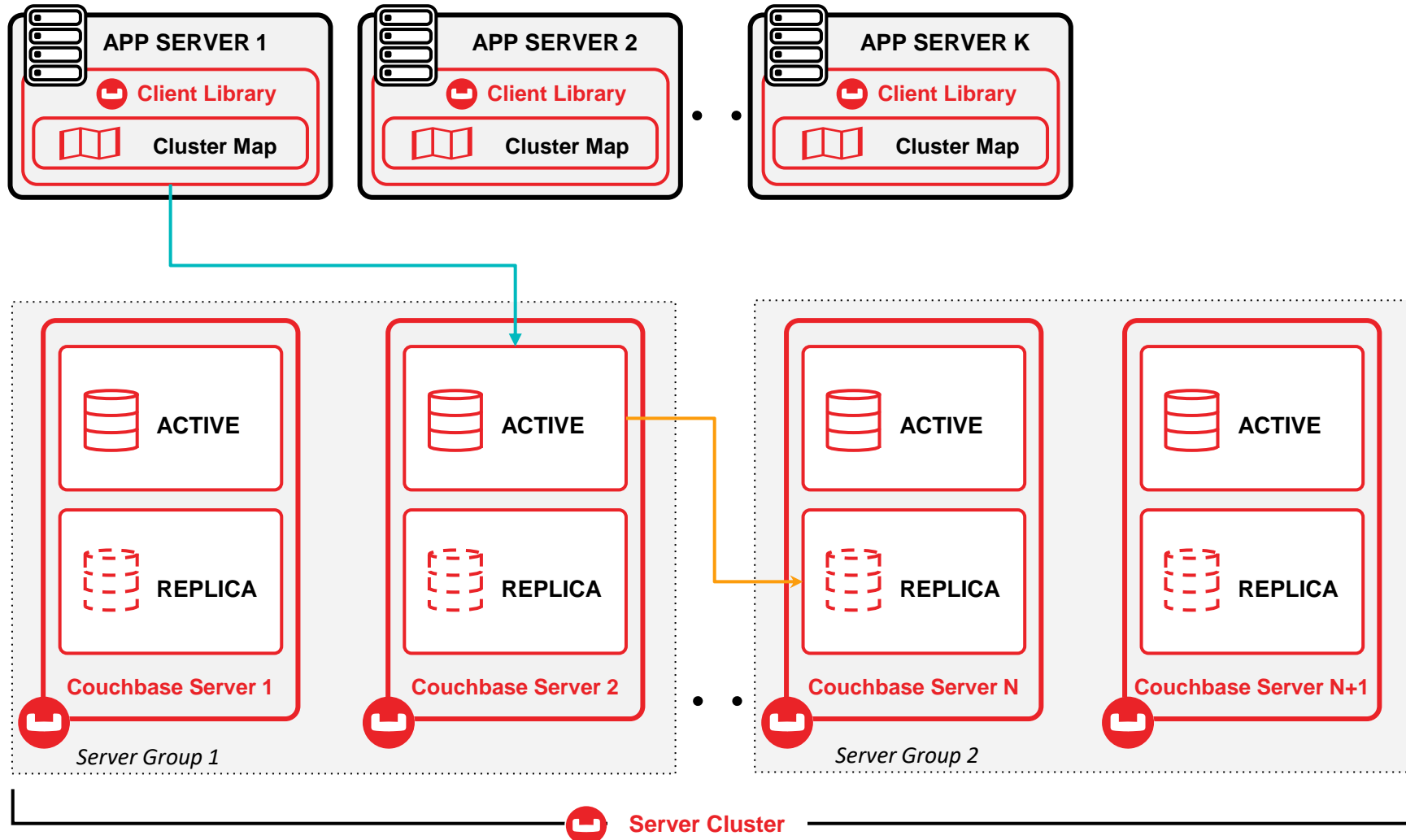
## Transaction

```
START TRANSACTION;
UPDATE cust SET b=b-1000
WHERE user="User1";
UPDATE cust SET b=b+1000
WHERE user="User2";
COMMIT ;
```





# 자동 샤딩, 복제 및 Fail-over



- 메모리-to-메모리 복제와 자동 Fail-over기능으로 “항상 가용한” 어플리케이션 운영 지원
- 자동 파티셔닝과 재분배가 보장되는 온-디맨드식 확장
- 고도의 데이터 보호 정책 : 3벌의 복제본까지 구성 선택 가능

# Full Stack 시큐리티

