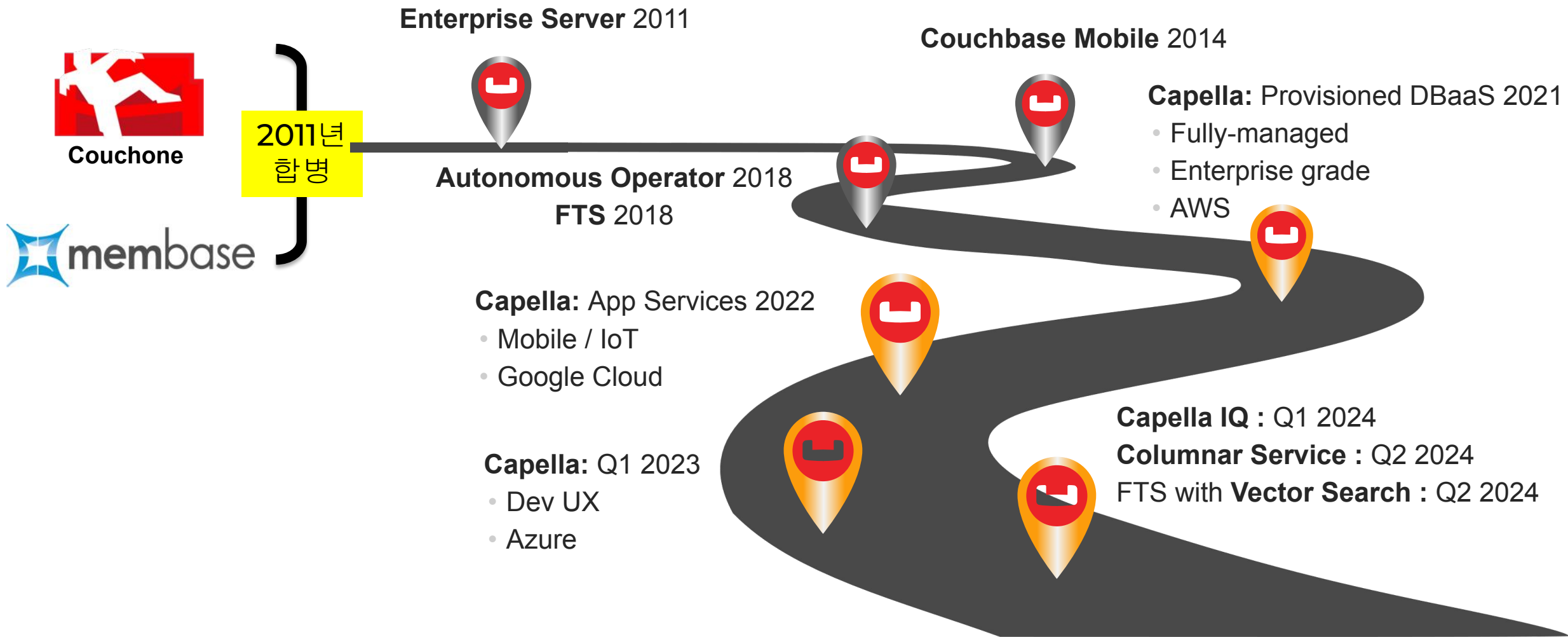


# 카우치베이스, 모던 어플리케이션을 위한 데이터 플랫폼

손광락, Solutions Engineer, 카우치베이스 한국

소개	1
아키텍처	2
데이터 모델	3
서비스(기능)	4
설치배포	5
Modern AI Powered 어플리케이션	6
개발지원	7

# Couchbase



# Couchbase 주요 고객

- 웹, 모바일, IoT 등 다양한 업무 환경에서 Couchbase를 도입하여 활용

Customers					
Application	캐싱 & 싱글뷰를 위한 세션 스토어	리얼 타임 프라이싱, 제품 캐탈로그, 재고관리	비행편 가용성, 예약, 가격분석등	Customer 360 싱글뷰, 'Unified notes'App지원	리얼타임 승무원 분석, 일정 및 리소스 관리
Performance	5백만+ 읽기/초당	1천만+ SKUs	8백만 Ops / 초당	2.1억개 다큐먼트	4.1만 종업원
	1천만 쿼리/초당	3만5천 요청/초당.	<2.5ms 반응시간	10만 사용자	1.5억+ 이용객





















































3 of the Top 10 eCommerce Companies

3 of the Top 3 GDS Companies

3 of the Top 3 Credit Reporting Companies

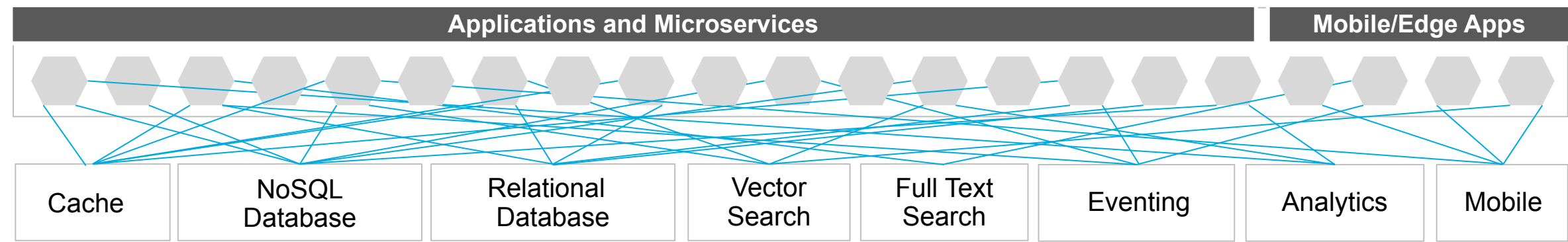
3 Fortune 500 Healthcare Companies

6 of the Top 10 Broadcast Companies

6 of the Top 10 Online Casino Gaming Companies

2 of the Top 2 IoT Platforms

# 우리의 현실 : Data Sprawl & Management Challenges



## Different Management and Security Systems

**Separate platform with multiple interfaces**

1. Independent deployment and management
2. Different data model and programming interfaces
3. Integration between multiple products
4. Support tickets with different vendors

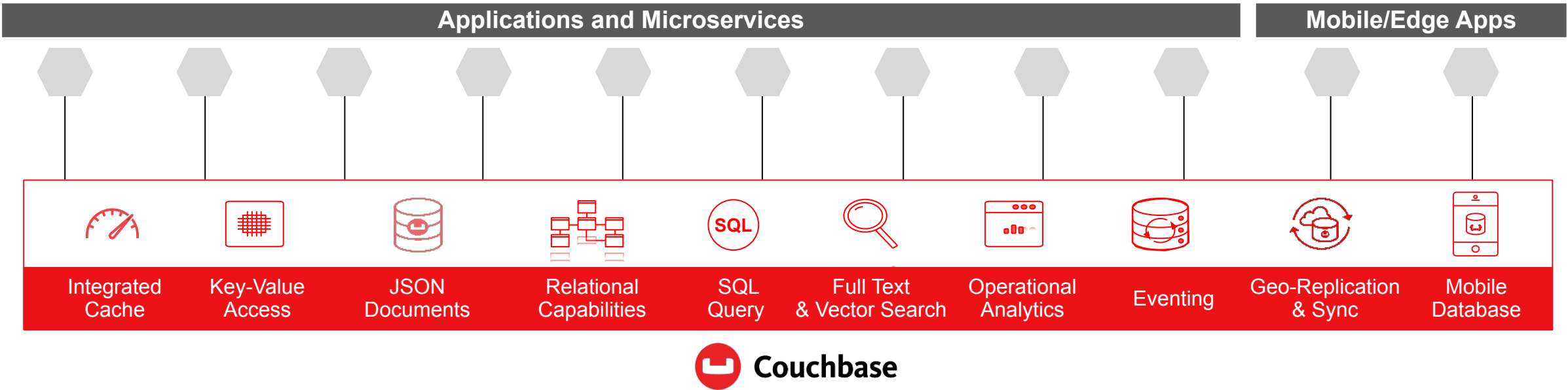
**Per product factors (Financial, time, & effort)**

1. License & agreement
  - Sourcing for renewals
  - Legal for agreements
2. Training
  - Developers
  - Operations
3. Support
4. Build API or connector to database
5. Purchase infrastructure

**COST**

- Infrastructure
- Licenses
- Integration
- Training
- Operational
- Support costs

# Couchbase 차별화, 데이터 서비스 플랫폼



## 빠르고

- **Memory-first design**
- Cloud-native scale
- Geo-replication via XDCR
- HA, DR & backup
- Low latency Cloud to Edge

## 유연하며

- **JSON document**
- Multimodel services
- Cloud deploy anywhere
- Mobile & Edge ready

## 익숙하게

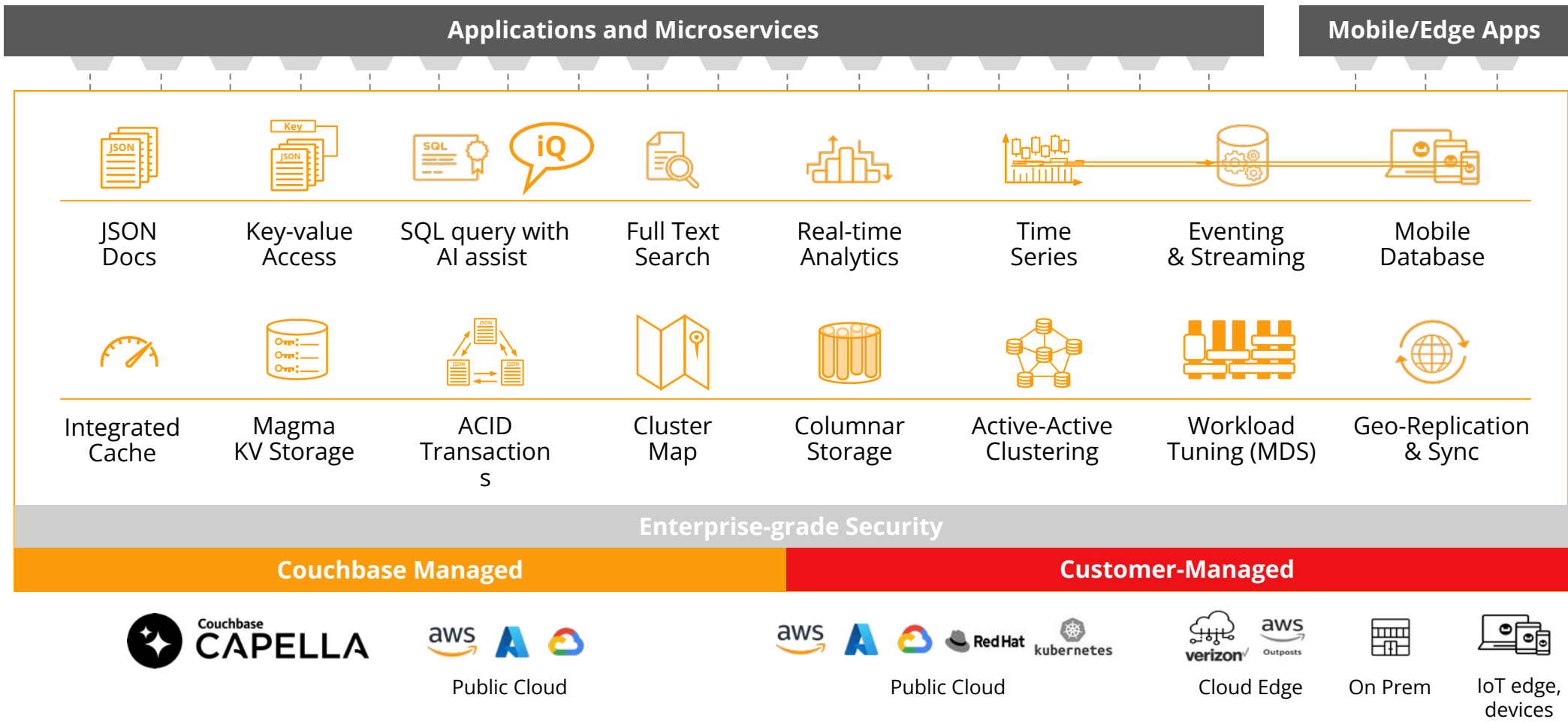
- **SQL++ query language**
- Dynamic Schema
- ACID SQL Transactions
- Cost-based optimizer
- SDKs for 12+ languages

## 합리적 투자

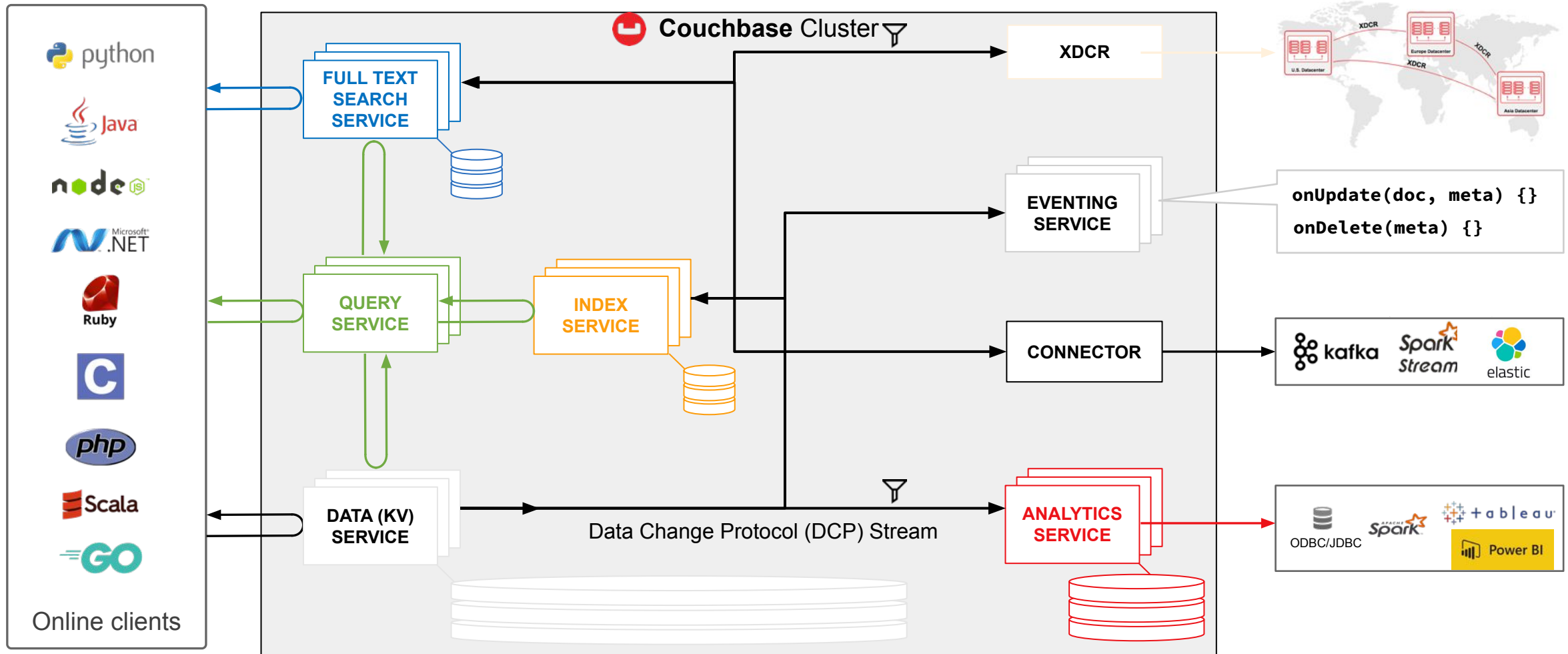
- Elastic scaling, sharding & rebalancing
- Multidimensional scaling
- High-density storage
- Incredible price/performance

# Enterprise 데이터 플랫폼

- JSON 도큐먼트 DB, Key-Value 캐시, 표준 SQL, 텍스트 검색, 실시간 분석, 시계열 처리, 고가용성, 재해복구, 모바일 DB 지원하며 **AI-Powered** 어플리케이션을 위한 데이터 플랫폼입니다.



# Couchbase 내부 구조



# Memory First 아키텍처

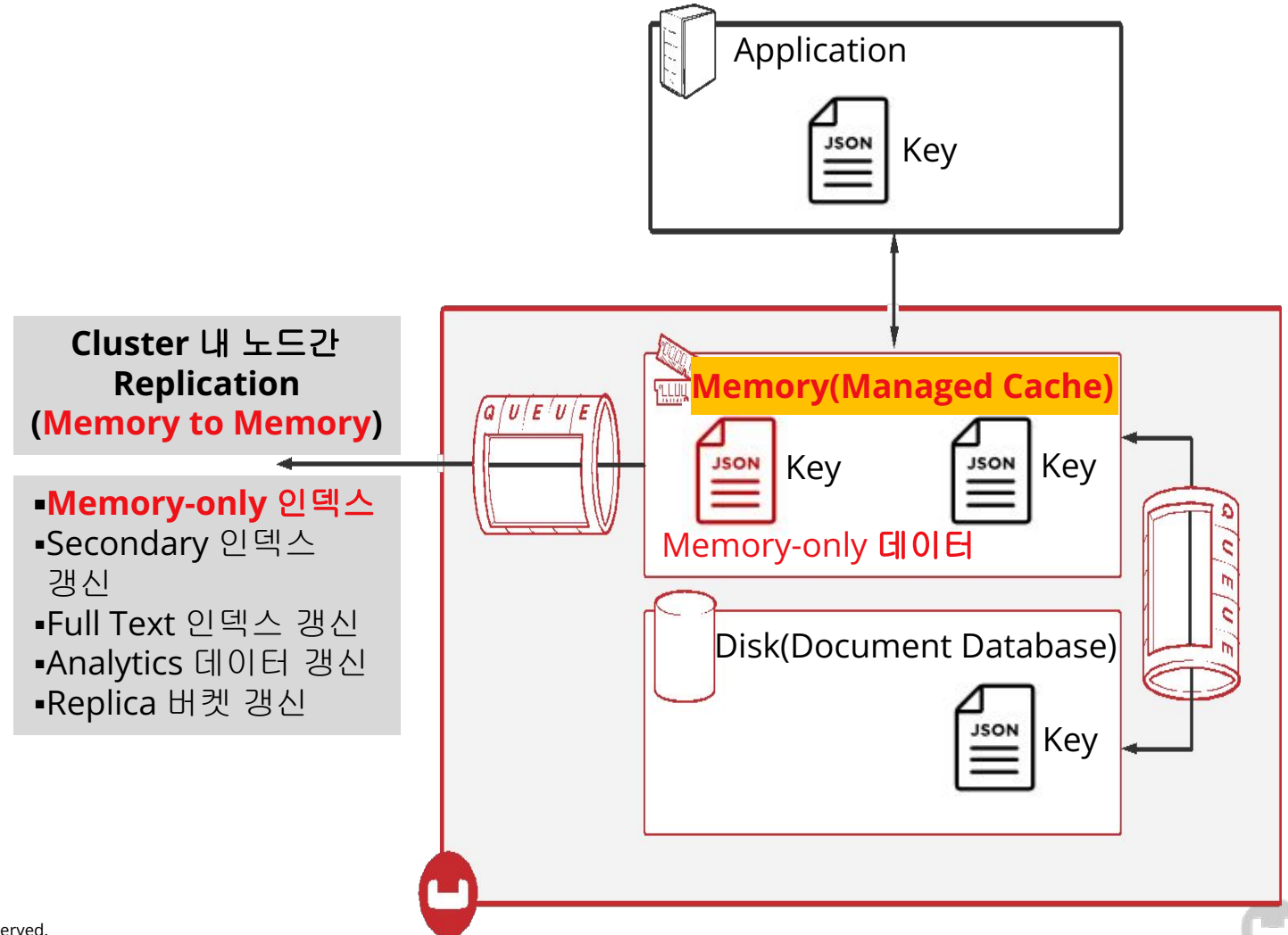
- 인 메모리 빌트인 캐시를 통해 빠른 Read/Write 업무를 수행하고 데이터 분산 관련 작업도 메모리 기반 프로토콜 사용

- 인메모리 **Key-Value** 오퍼레이션

- 특정 **Key**를 기준으로 데이터를 인메모리에서 처리하는 메카니즘
- 대부분 도큐먼트 데이터베이스는 Read 성능 향상을 위해 별도 솔루션으로 적용

- Couchbase**

- 인메모리 **Key-Value** 오퍼레이션의 장점을 구현한 빌트인 캐시 제공
- Value**가 단순 수치나 배열이 아닌 **JSON** 도큐먼트 자체
- JSON** 도큐먼트 처리가 메모리 우선 방식





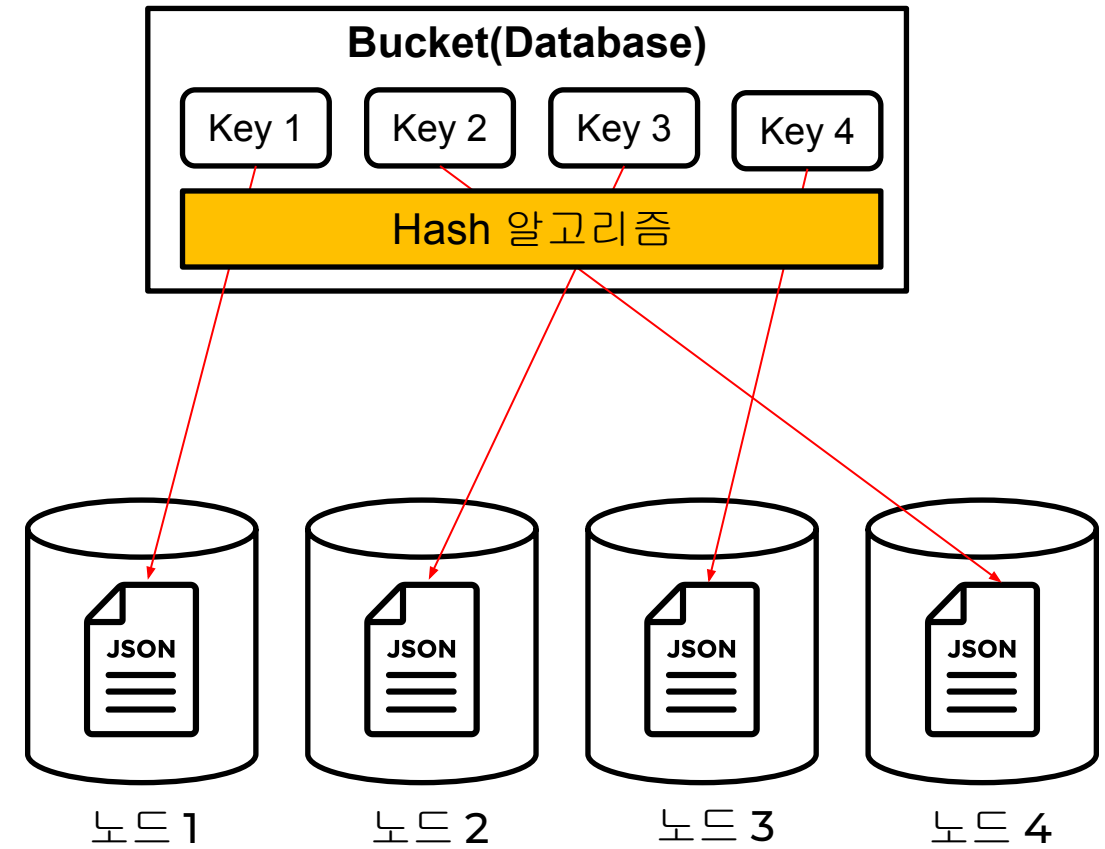
# 분산 병렬 데이터베이스

## • Key 기반 자동 분산 아키텍처

- 대량의 Key-Value 처리를 노드 별로 분산하여 성능 향상
- 최대한 균등하게 분산 저장 가능, 특정 노드에 편중되는 현상 (Data Skew) 방지
- 별도의 분산 정책 불필요

## • Couchbase

- Key에 대한 Hash 알고리즘 적용으로 자동 분산
- 노드 추가 시, Rebalancing을 통해 Key 재 분산 수행



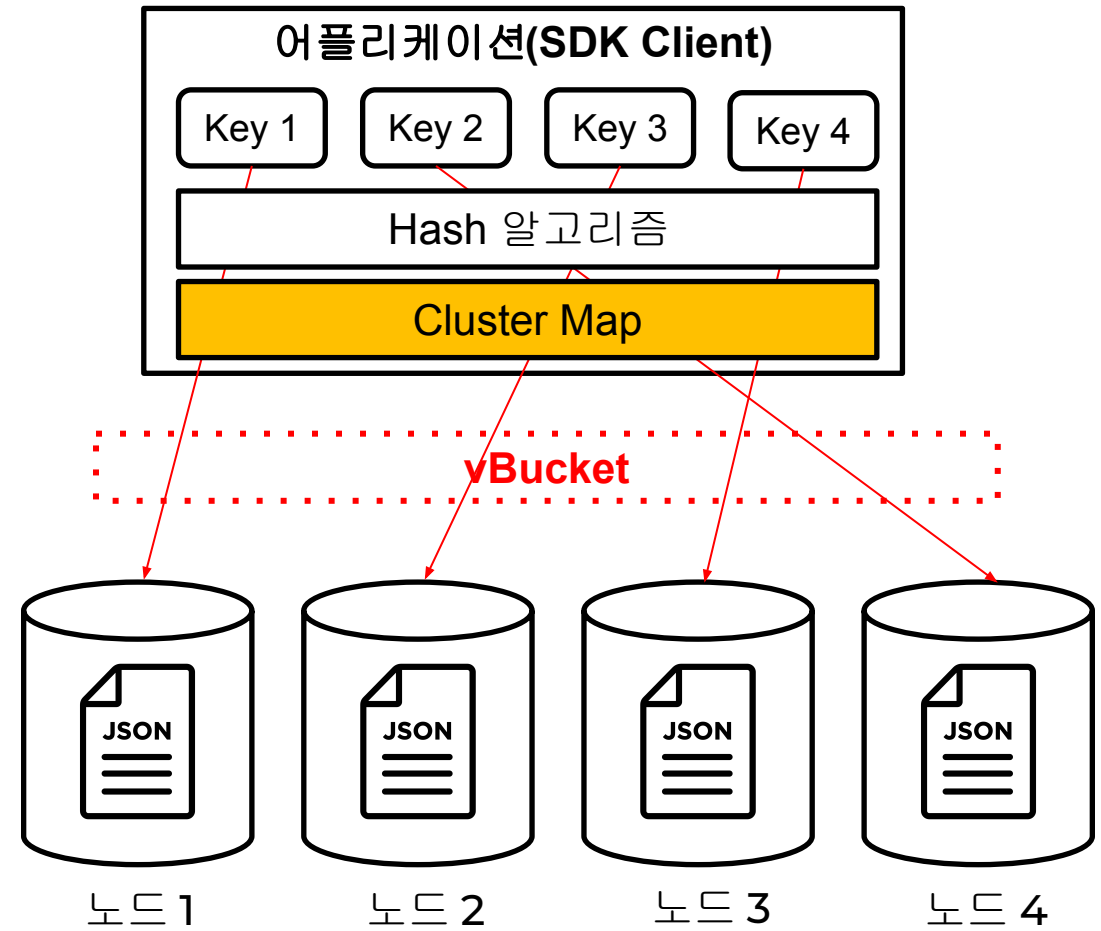
# Master Node-less 아키텍처

- **Master Node-less 아키텍처**

- 어플리케이션의 Key-Value Operation 시, 해당 Key에 매핑된 특정 노드에 직접 접근
- 모든 노드가 어플리케이션 측면에서 **Active** 노드 역할

- **Couchbase**

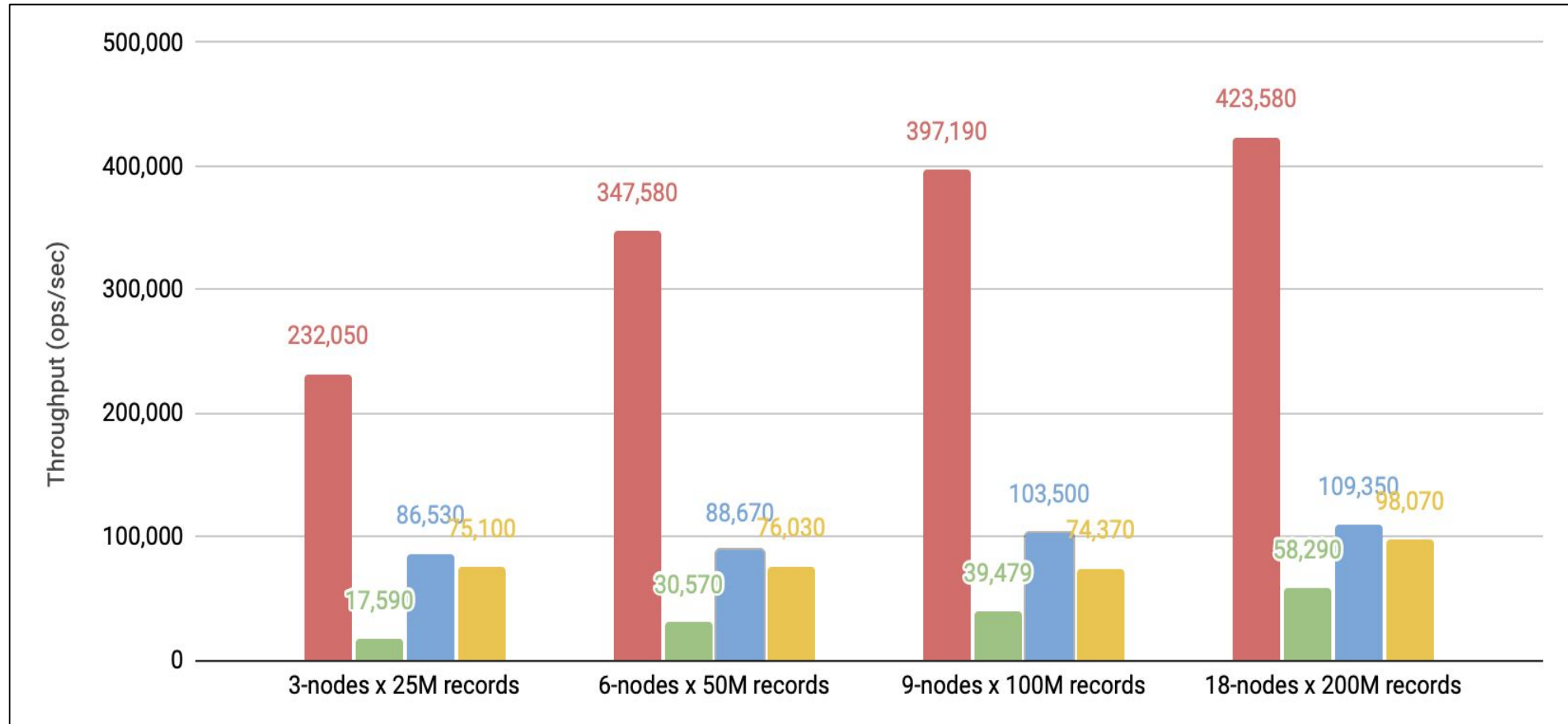
- 어플리케이션이 Couchbase에 데이터 처리를 구현하기 위해 SDK 활용
- 데이터 분산 정보(Cluster Map)를 지속적으로 SDK Client에 Update



# 차별화된 데이터 처리 성능

- 다른 NoSQL 기반 데이터베이스 대비 최소 4 배 이상의 데이터 처리 능력

Couchbase Capella | MongoDB Atlas | AWS DynamoDB | Redis Enterprise Cloud



# 성능 Demo

Cluster > Buckets

activity 1 help admin

Dashboard  
Servers  
Buckets  
XDCR  
Settings  
Documents  
Query  
Indexes  
Search  
Analytics  
Eventing  
Views

filter buckets...

name	items	resident	ops/sec	RAM used/quota	disk used	
data	21,345,902	100%	406,160.4	11.2GiB / 1.14TiB	27.8GiB	Documents Scopes & Collections
gen	10,000	100%	0	103MiB / 5.85GiB	96.9MiB	Documents Scopes & Collections
meta	2,048	100%	102.2	93.2MiB / 11.7GiB	84.5MiB	Documents Scopes & Collections
start	1	100%	0	89.5MiB / 5.85GiB	19MiB	Documents Scopes & Collections

1 초당 400,000 건 이상의 Write 성능

Node 별 스펙

- Data Node 6대 : 32 core/256 GB mem/500 GB Disk

- Eventing Node : 16 core/32 GB mem/50 GB Disk

00:01:02.20

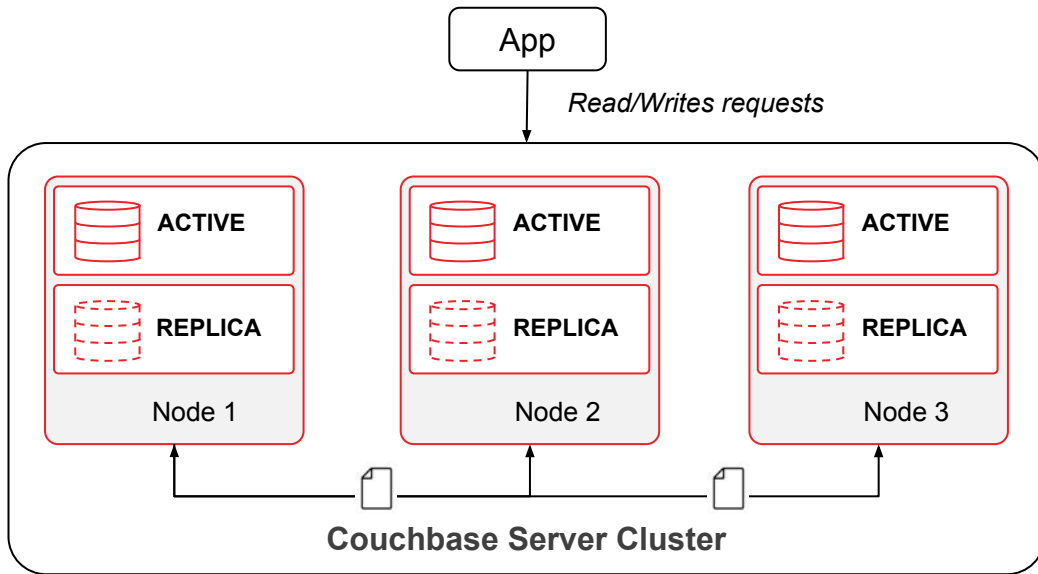
hr min sec

⏸ ⏮

# 가용성 : 분산 환경에서 시스템 안정성

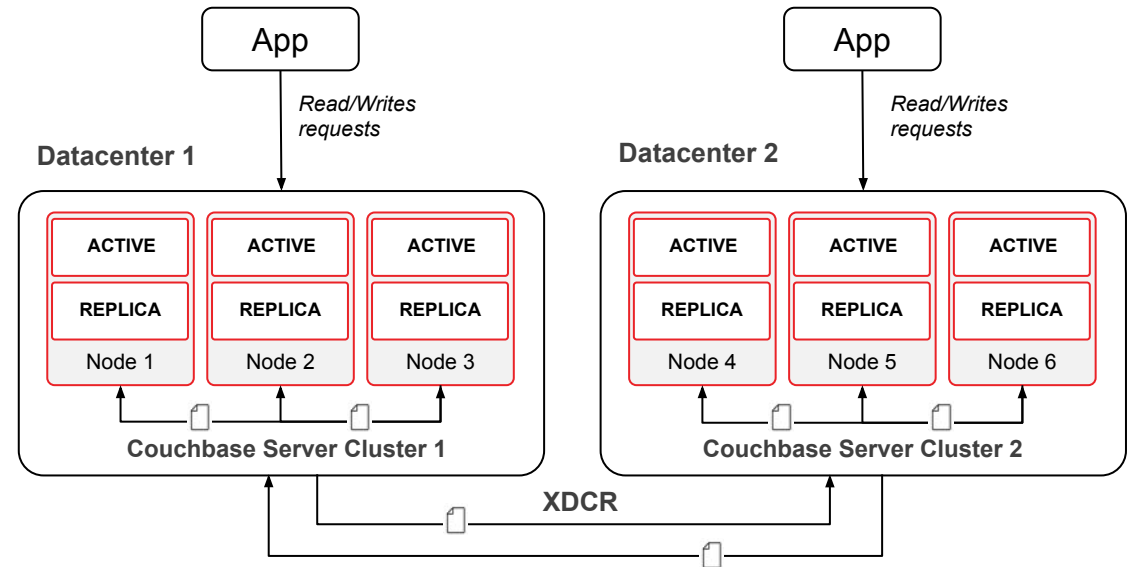
- 빠른 Fail-over를 통한 고가용성 구현 및 클러스터 간 복제 기능을 통한 재해복구 솔루션 제공

## 고가용성 : Intra Cluster Replication



- Auto-sharding을 통한 균등한 데이터 분산
- Active shard의 데이터 변경 내역은 지속적으로 Replica shard로 복제
- 노드 장애가 발생하면 Replica shard가 Active로 전환

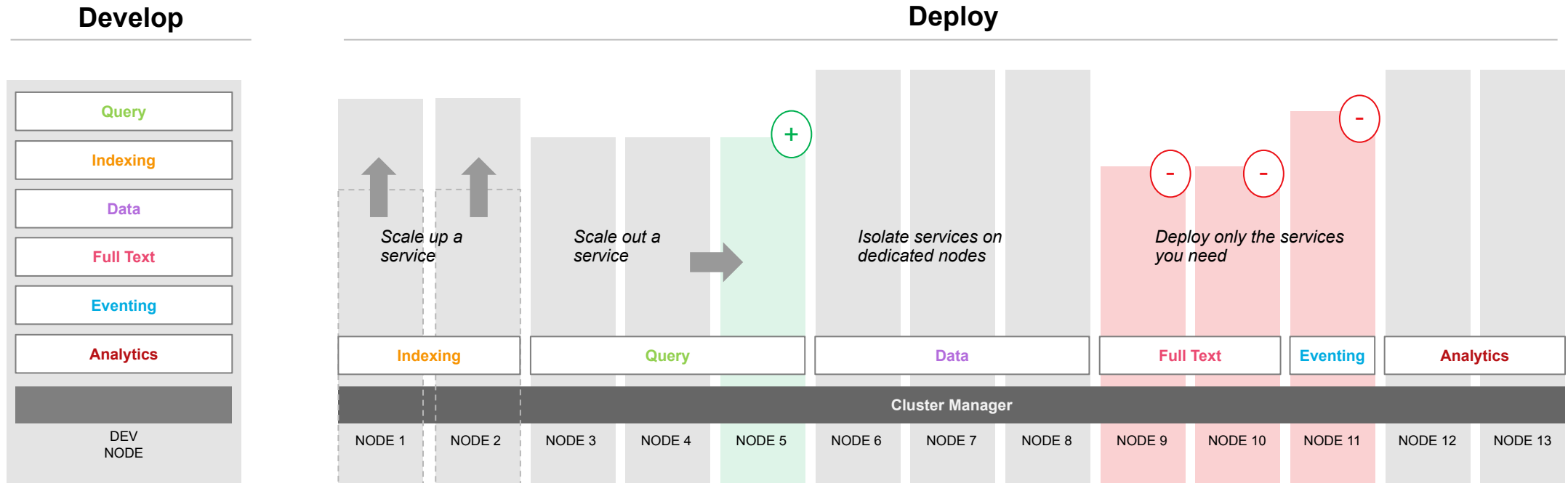
## 재해 복구 : Cross Data Center Replication



- XDCR을 통해 멀리 데이터 센터에 위치한 클러스터 간 데이터 복제
- 단방향 복제 및 양방향 복제 지원
- 복제는 필요한 데이터만 필터링 가능

# 확장성 : MDS(Multi-Dimensional Scaling)

- Couchbase는 Multi-Dimensional Scaling 기능으로 서비스 별 Workload 분산 및 독립성을 보장



## Service 단위 하드웨어 자원 최적화

각 Service 별 시스템 자원을 독립적으로 할당

각 **Service**에서 수행되는 작업이 다른 **Service**에 영향을 최소화, 예를 들어 Analytics Service에서 복잡한 작업을 수행하여 시스템 자원을 많이 사용해도 Data Service 혹은 Query Service에서 수행하는 Operational 작업에는 영향이 없는 구조

# 운영편의성 : 백업복구

- 자체 백업 / 복구 기능이 있어, 별도 백업 솔루션이 필요 없음

## • Backup Service

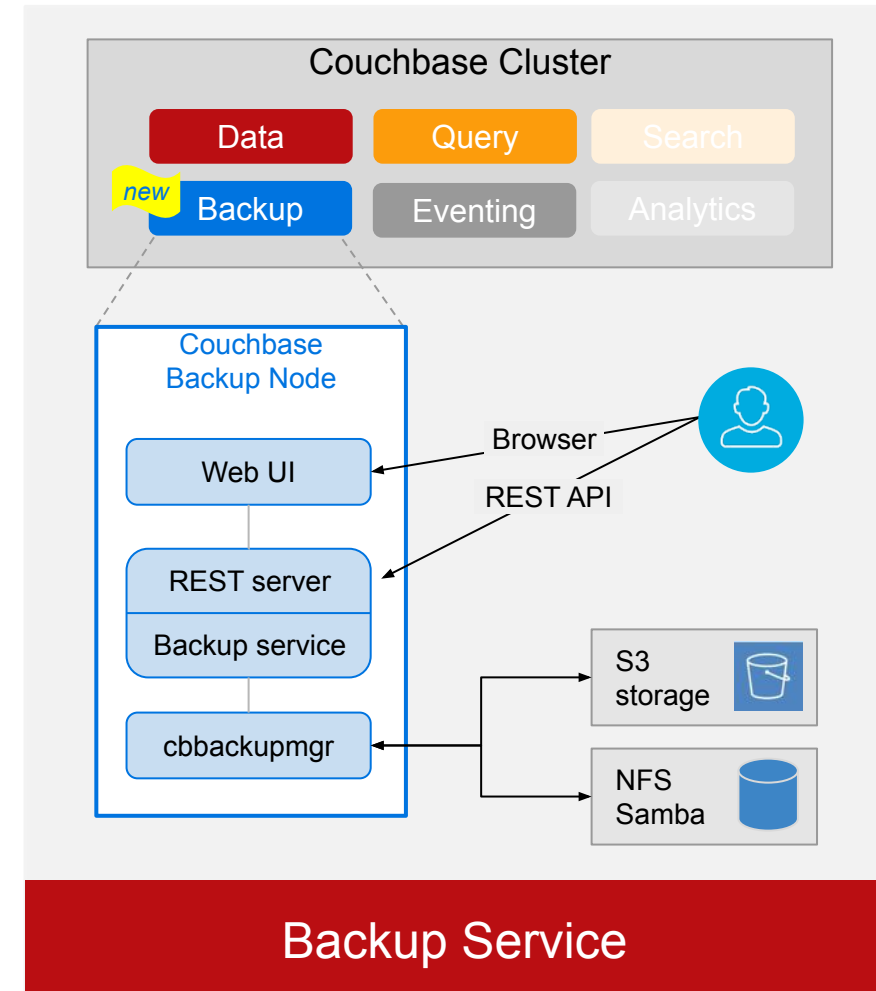
- UI 기반 Backup & Recovery
- 공유 파일 시스템 구성 필수
- Backup Scheduler 제공
- 백업 중 장애를 위한 Resume 기능 제공
- 병렬 백업

## • Backup 방식

- Full Backup
- Incremental Backup
- Merge Backup File

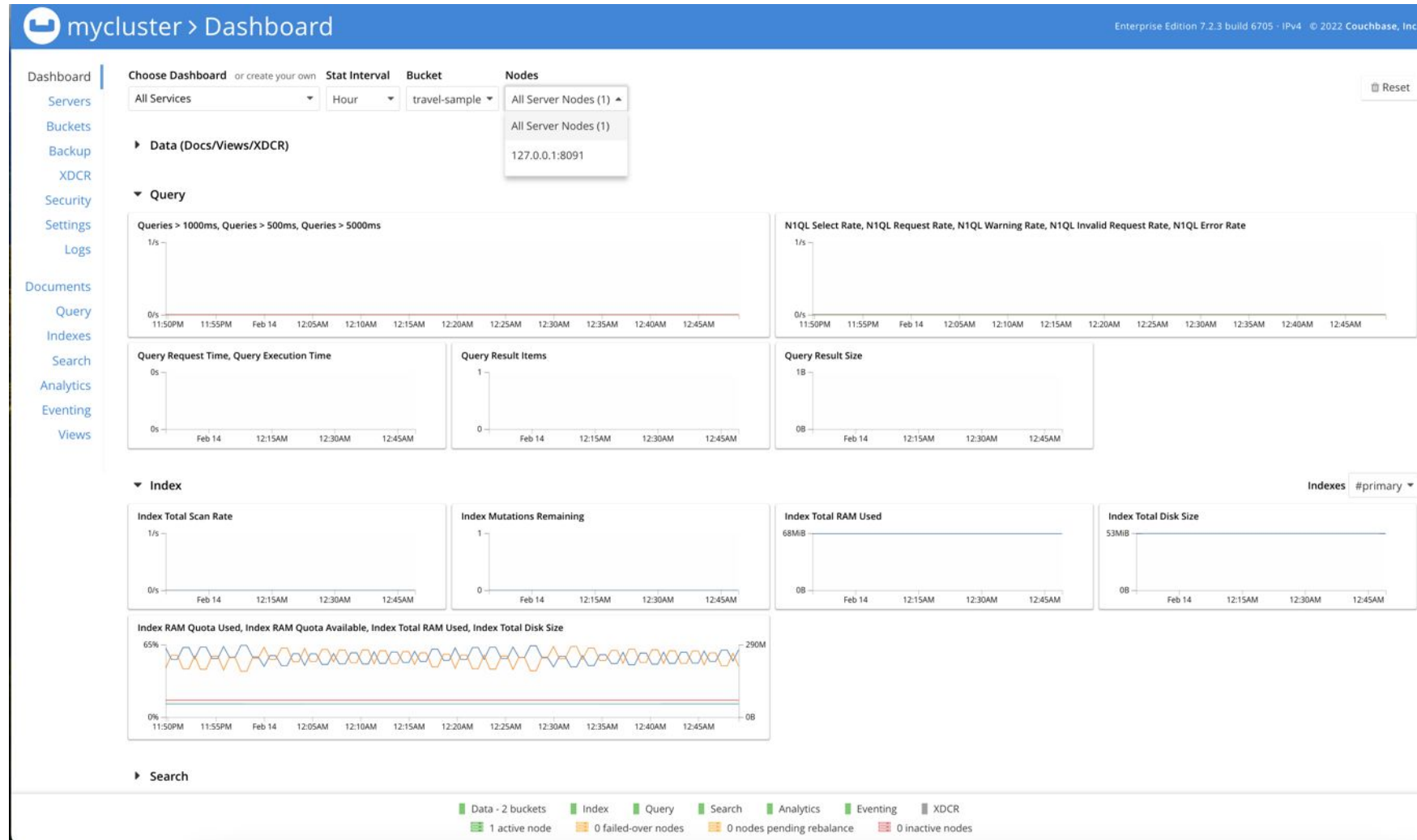
## • Restore 방식

- Bucket, Scope, Collection 단위 가능



# 운영편의성 : 성능 모니터링

- Couchbase는 다양한 성능 매트릭을 제공하며, Prometheus/Grafana 로 관리 가능





# 운영편의성 : 시스템 상태 경고 및 권고

- Couchbase는 매트릭이 임계 수준을 넘어서면 경고(Alerts) 보내게 되고, 권고 사항(Recommended actions)도 제시됩니다. 경고 수준(Severity Level)은 Info, Warning, Critical 이 있음

mycluster > Settings

GeneralAuto-CompactionAlertsSample Buckets

Dashboard

Servers

Buckets

Backup

XDCR

Security

Settings

Logs

Documents

Query

Indexes

Search

Analytics

Eventing

Views

Enable email alerts

Email Server Host

localhost

Port

25

Username

Password

☐ Require encryption (TLS)

Sender Email

the FROM field

couchbase@localhost

Recipients

separate addresses with comma "," or spaces " "


root@localhost

Send Test Email

Available Alerts

<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Node was auto-failed-over
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Maximum number of auto-failed-over nodes was reached
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Node was not auto-failed-over as other nodes are down at the same time
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Node was not auto-failed-over as there are not enough nodes in the cluster running the same service
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Node was not auto-failed-over as auto-failover for one or more services running on the node is disabled
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Node's IP address has changed unexpectedly
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Disk space used for persistent storage has reached at least 90% of capacity
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Metadata overhead is more than 50%
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Bucket memory on a node is entirely used for metadata
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Writing data to disk for a specific bucket has failed
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Writing event to audit log has failed
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Approaching full Indexer RAM warning
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Remote mutation timestamp exceeded drift threshold
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Communication issues among some nodes in the cluster
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Node's time is out of sync with some nodes in the cluster
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Disk usage analyzer is stuck; cannot fetch disk usage data
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Memory usage threshold exceeded
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	History size threshold exceeded
<input checked="" type="checkbox"/> Email	<input checked="" type="checkbox"/> UI popup	Approaching Indexer low resident percentage

Confidential and Proprietary



17

# JSON 도큐먼트 데이터베이스

- JSON 도큐먼트 기반의 데이터 입/출력 처리로 유연한 데이터 모델을 구현하고 개발 주기를 단순하고 신속하게 지원

이 력 서

1. 기초자료

성명	홍○○○한자	원○○○
주민등록번호	000000-0000000	
E-mail	bizforms@bizforms.co.kr	
전화번호	02-000-0000	휴대폰 000-123-4567
우편번호	○○○○○○○○	주소
주	○○○○○○○○	○○○○○○○○
본	○○○○○○○○	○○○○○○○○
호적관계	호주	관계장남

기초정보

2. 학력사항

년/월/일	학	교	명	비	고
20○○.02	○○정보	공업고등학교	졸업		

학력사항

경력사항

개인능력

관련내용	비고
리눅스(레드햇기반)운영	
samba서버 운영(nfs기반)	
apache웹서버 운영	
sendmail서버 운영	
shell 스크립트(bash) 작성 가능	
perl 및 C도 기본적으로 이해 가능	
DNS서버 운영	
WINDOWS2000 서버 운영 및 BS, AD(분산파일시스템), DNS	
NAT, DHCP, SMTP, POP3 등의 service 운영가능	
전반적인 네트워크구성 이해(TCP/IP기본)	

JSON 도큐먼트

```
KEY : 1001
{
  "성명": "홍길동",
  "주소": "서울시 00구 00동 000-000",
  "E-mail": "HongKildong@couchbase.com",
  ...
  "학력사항": [
    {
      "졸업년도": "2019년",
      "학교명": "00정보 공업고등학교"
    }
  ],
  "경력사항": [
    {
      "기간": "2019 ~ 현재",
      "관련내용": "XX글로벌 IT팀 Unix 서버
담당"
    }
  ],
  "개인능력": [
    {
      "관련내용": "리눅스 운영"
      "비고": NULL
    }
  ],
  ...
}
```

## JSON 도큐먼트의 장점

- 단일 도큐먼트 내에 다양한 정보를 계층 구조를 활용하여 저장
- 정보 추가/삭제가 유연한 구조 제공
- 데이터 전달을 위한 표준 인터페이스 역할

## RDB와 차별점

- 여러 테이블로 분리, 저장되는 데이터를 단일 도큐먼트에 저장
- 테이블 간 조인을 최소화하여 데이터 처리 속도 향상

# 표준 SQL 지원

- 표준 SQL 문법을 통해 개발자의 기존 기술 능력을 활용하고 학습 시간 단축



```
SELECT
  c.name,
  a.title,
  a.actduration,
  a.startDate,
  SUM(a.actduration) OVER ( PARTITION BY
    c.name ORDER BY c.name, a.startDate ) running_total,
  TRUNC(100*(a.actduration/SUM(a.actduration)
    OVER(PARTITION BY c.name)) ) pct_of_total_time,
  RANK() OVER(PARTITION BY c.name ORDER BY
    (ARRAY_COUNT(a.contacts)/
    ARRAY_COUNT(c.contacts)) DESC) hightouch_rank
FROM activity a
  INNER JOIN account c
    ON (a.accid = c.id)
WHERE a.activityType = 'Appointment'
  AND a.startDate BETWEEN '2018-10' AND '2018-12'
GROUP BY c.name, a.title, a.startDate,
  a.actduration, a.contacts, c.contacts
ORDER BY c.name, a.startDate
```

21 lines vs 347 lines



```
db.activity.aggregate([
  { $match : { type: "activity" } },
  { $match : { activityType: "Appointment" } },
  { $match : { startDate: { $gt: '2018-10-01',
    $lt: '2018-12-31' } } },
  { $lookup: {
    from: "account",
    localField: "accid",
    foreignField: "id",
    as: "account_docs" },
  { $match : { "account_docs": { $ne: [] } } },
  { $unwind: "$account_docs" },
  { $group : { "_id": {
    name: "$account_docs.name",
    title: "$title",
    startDate: "$startDate",
    duration: "$actduration",
    activity_contacts: "$contacts",
    account_contacts: "$account_docs.contacts",
  } } },
  { $addFields: { total_time: total_time } },
  { $addFields: { hightouch_rank: rank_temp } },
  { $addFields: { running_total: running_total } },
  { $project: {
    "_id": 0,
    name: "$_id.name",
    title: "$_id.title",
    startDate: "$_id.startDate",
    duration: "$_id.duration",
    activity_contacts:
      "$_id.activity_contacts",
    account_contacts: .....
  } } ])
```

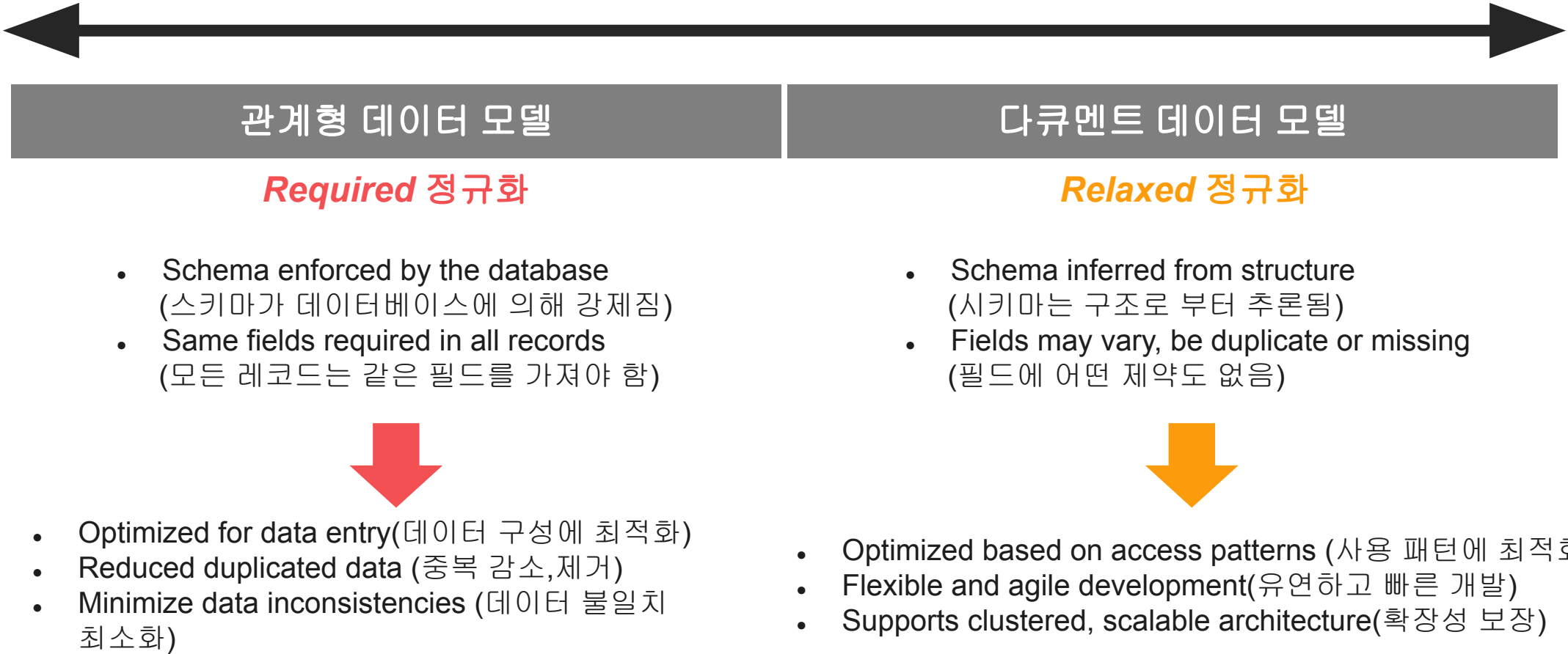
# 논리 / 물리 모델

- RDBMS와 유사한 구조의 논리 계층 구조로 구성하여 편리한 데이터 관리
- Data 서비스를 완전 메모리DB로 사용도 가능하며 용도에 따라 물리 저장 방식을 선택할 수 있음

RDBMS	Couchbase
Server	Cluster
Database	Bucket
Schema	Scope
Table	Collection
Row	Document (JSON)
Value	Sub-Document, Array

Feature	Ephemeral Bucket	Couchbase Bucket	Magma Bucket
Bucket memory quota (per node)	Min 256MB	Min 256MB	Min 1024MB
Max Object Size	20MB	20MB	20MB
Persistence	no	yes	yes
Replication and XDCR	yes	yes	yes
Encrypted data access	yes	yes	yes
Rebalance	yes	yes	yes
N1QL, Seach, Analytics, Eventing	yes	yes	yes
Indexing	yes	yes	yes
Backup	yes	yes	yes

# 관계형 vs. 다큐멘터 데이터 모델



# Real World Data 를 JSON으로 표현

Customer		
CustomerID	Name	DoB
CBL2017	Jane Smith	1990-01-30

Customer Document Key: CBL2017

```
{  
  "Name" : "Jane Smith",  
  "DOB"  : "1990-01-30"  
}
```

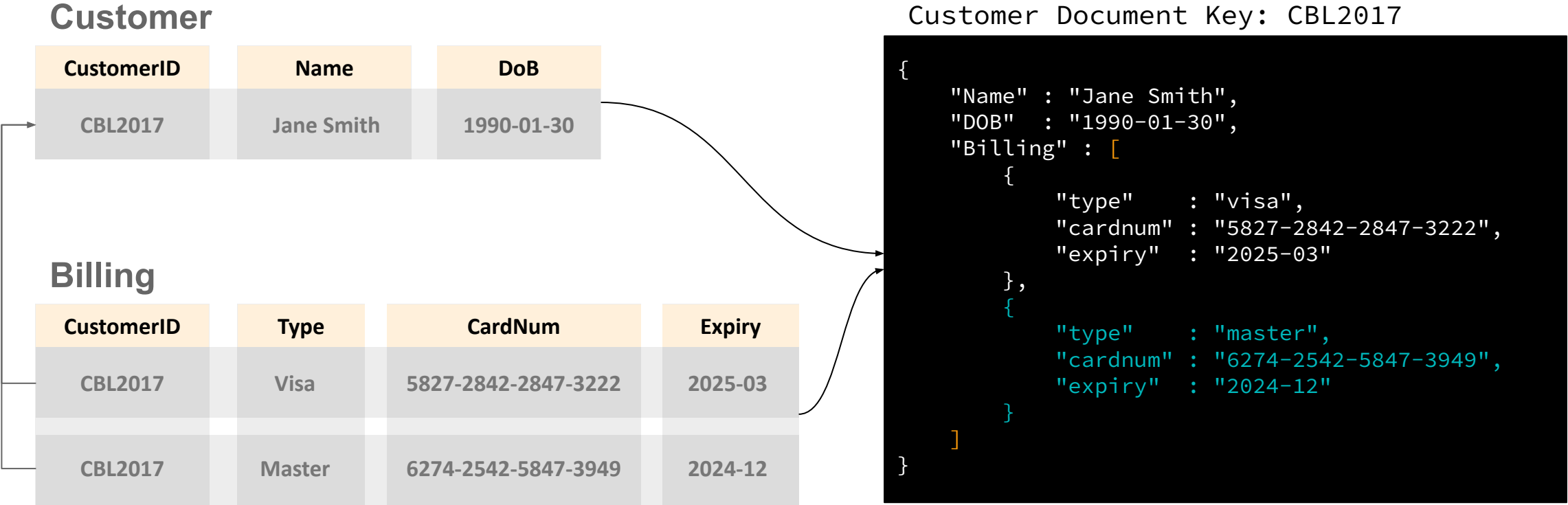
OR

Customer Document Key: CBL2017

```
{  
  "Name" : {  
    "fname": "Jane",  
    "lname": "Smith"  
  },  
  "DOB"  : "1990-01-30"  
}
```

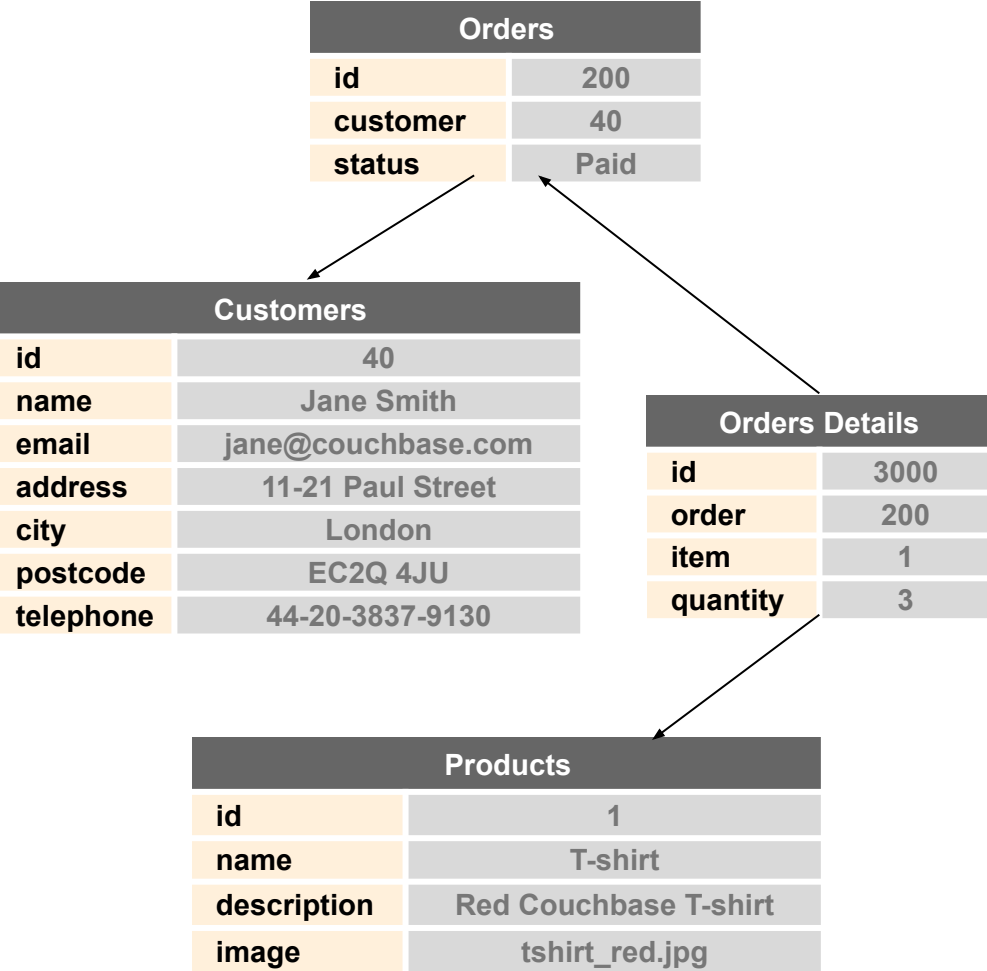
- The Primary Key becomes the Document Key
- Column name-Column value become KEY-VALUE pair

# 고객의 카드 정보를 JSON으로 표현



- **Denormalization simplifies data access and offers the best performance**  
비정규화는 데이터 사용을 단순화 해주고, 최고의 성능을 제공함
- **Value evolution: Simply add additional array element or update a value**

# 테이블(RDBMS)을 컬렉션(NoSQL)에 매핑 | eCommerce 예제



Bucket: ecommerce | Scope:\_default

Collection: 'Products'

```
{
  "name": "T-shirt",
  "description": "Red Couchbase T-shirt",
  "image": "tshirt_red.jpg"
}
```

DocID:1

Collection: 'Customers'

```
{
  "name": "Jane Smith",
  "email": "jane@couchbase.com",
  "address": "11-21 Paul Street",
  "city": "London",
  "postCode": "EC2A 4JU",
  "telephone": "44-20-3837-9130"
}
```

DocID:40

Collection: 'Orders'

```
{
  "customer": {
    "id": 40,
    "name": "Jane smith",
    "email": "jane@couchbase.com"
  },
  "status": "Paid",
  "orderDetails": [
    { "productId": 1, "name": "T-shirt", "quantity": 3 },
  ]
}
```

DocID:200



# Trade-offs 극복 > 앱에 최적화

분산 시스템에서 데이터 모델을 수립할 때 장점과 단점이 상호 존재 함



## Document Size 문서크기

- Impact on disk space
- Impact on time to transfer across the network
- Impact on time to serialize/deserialize
- Impact on queries (e.g. JOIN slower than lookups)



## Atomicity 원자성

- Transactions provide ACID guarantees across multiple documents
- Transactions are slower than atomic KV operations at the document level



## Complexity 복잡성

- Simple data models are easier to maintain
- Complex data models lead to complex (and slower) queries and indexes
- Smaller collections increase indexing performances



## Speed 성능

- Performances depends on documents size, complexity of the queries and resources available - compute, storage and network
- Optimize data model wrt. the most used queries

# JSON 데이터 모델시 고려사항



## EMBED WHEN

- There is an Ownership Relationship  
오너쉽 관계가 명확할 때
- Both docs are frequently accessed together  
두 문서가 동시에 사용되는 경우가 많을 때
- Reads greatly outnumber writes  
읽기가 쓰기에 비해 많을 때
- Data is small  
데이터 사이즈가 작을 때



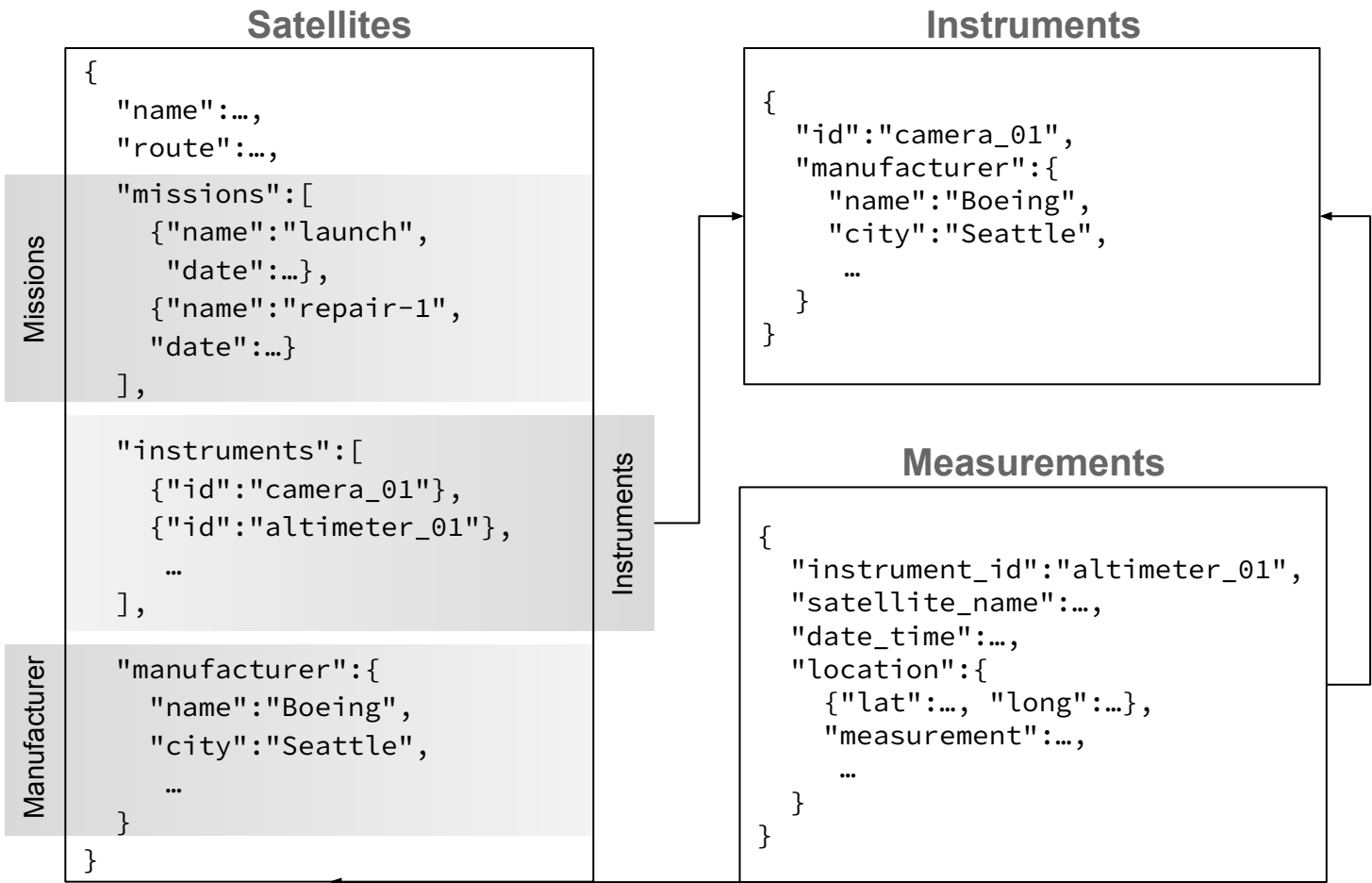
## REFER WHEN

- There is not an Ownership Relationship  
오너쉽 관계가 명확하지 않을 때
- Both docs are not frequently accessed together  
두 문서가 동시에 사용되는 경우가 적을 때
- Document is updated frequently  
문서가 자주 변경될 때
- Need to reduce the document size  
문서의 크기를 줄일 필요가 있을 때

Try to embed first, refer when it makes sense  
먼저 **Embed**를 시도해 보고, 그 다음 **Refer** 고려

# Relationships에 따른 Embed 와 Refer

- **1-1**  
**Embed** Example: Satellite and Manufacturer
- **1-Many**  
**Embed** Example: Satellite and Missions  
**Reference** Example: Measurements, Satellites and Instruments
- **Many-Many**  
**Reference** Example: Satellite and Instruments



# Time-Series 포맷 지원

- **Time Series Format**

- JSON Array를 사용하여 대용량 데이터를 효율적으로 저장
- ts\_start, ts\_end, ts\_interval, ts\_data 규격에 맞게 데이터 구조화 필요

- **데이터 활용**

- \_timeseries function을 통해 테이블 형태의 데이터로 전환 가능
- 인덱스 하나로 모든 Query 대응
- 일반 JSON 문서 대비 사이즈 축소를 통해 이력 데이터 용도로 활용

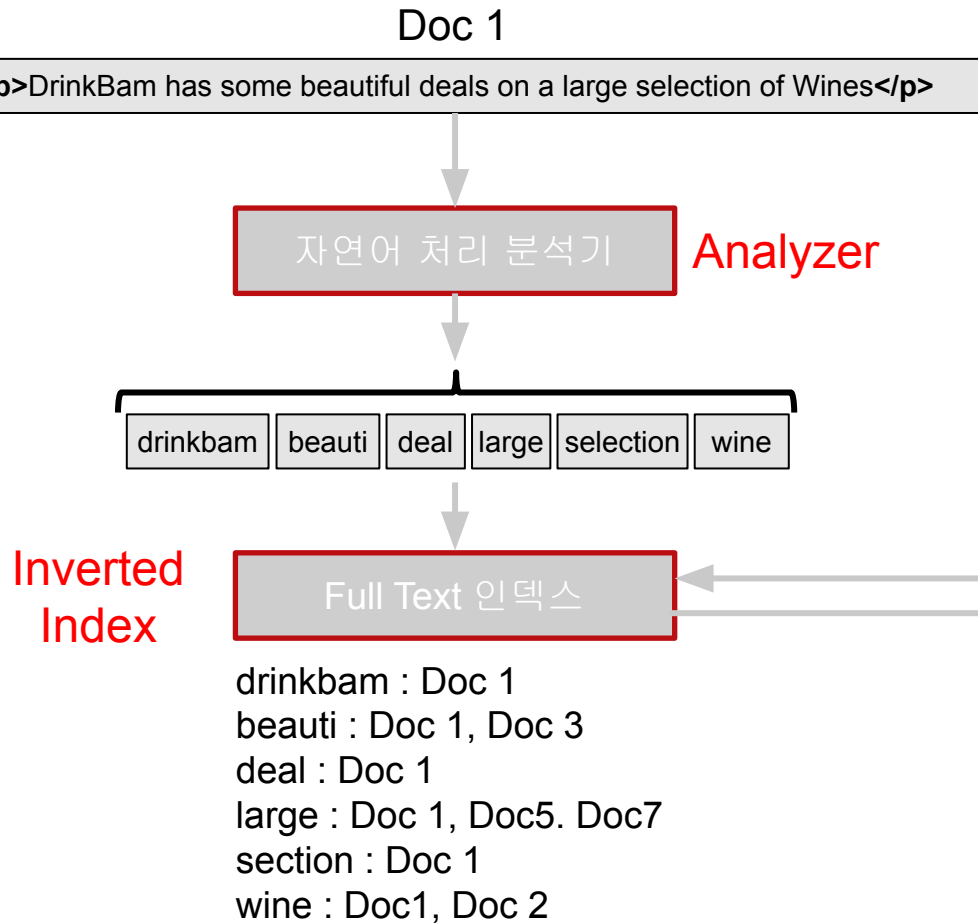
```
{
  "equip_id": "1",
  "lot_id": "1",
  "ts_start": 1375228800000,
  "ts_end": 1372636800000,
  "ts_interval": 100
  "ts_data": [
    [10, 27], [10, 27], [10, 27], [10, 27], [10, 30], [10, 30], [10, 30],
    [10, 30], [10, 30], [10, 30], [10, 30], [10, 30], [10, 30], [10, 30],
    [10, 30], [10, 30], [10, 30], [10, 30], [10, 30], [10, 30], [10, 30],
    [10, 23], [10, 23], [10, 23], [10, 23], [10, 23], [10, 23], [10, 23],
    [10, 23], [10, 23], [10, 23]
  ]
}
```

```
SELECT t.*
FROM Raw_Collection AS d
UNNEST _timeseries(d, {"ts_ranges":$ts_ranges}) AS t
WHERE d.equip_id = '1'
AND (d.ts_start <= $ts_ranges[1]
AND d.ts_end >= $ts_ranges[0]);
```

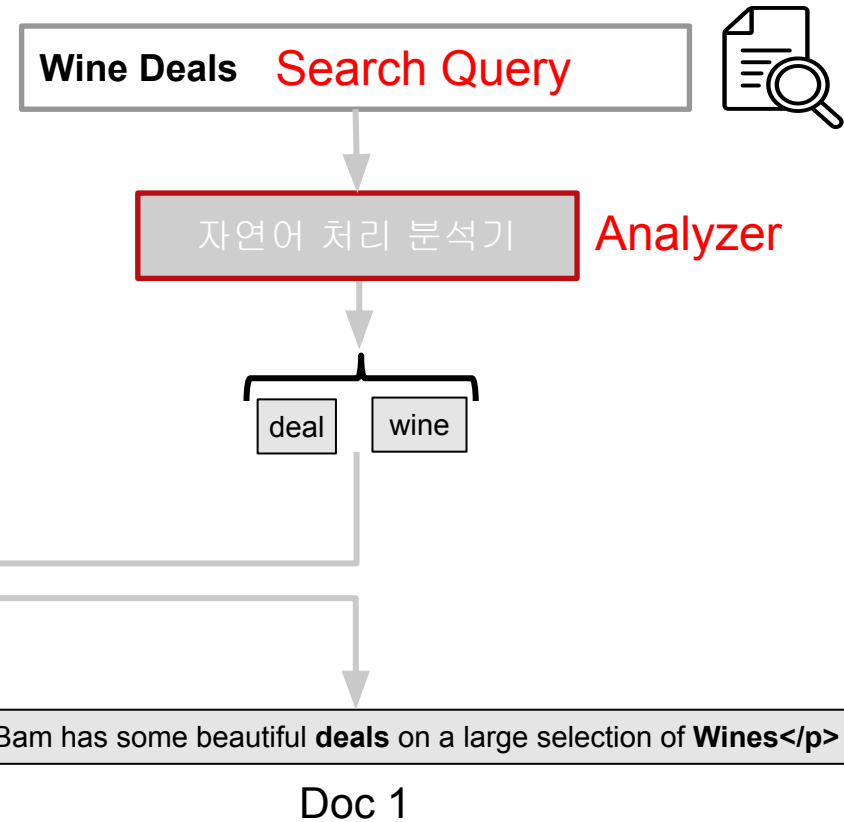
# Full Text Search Service

- Data와 Query 서비스에 통합하여 검색 서비스 제공

## <1. 검색을 위한 인덱스 구성>



## <2. 검색어를 통한 문서 검색>



# Eventing Service

## • 개요

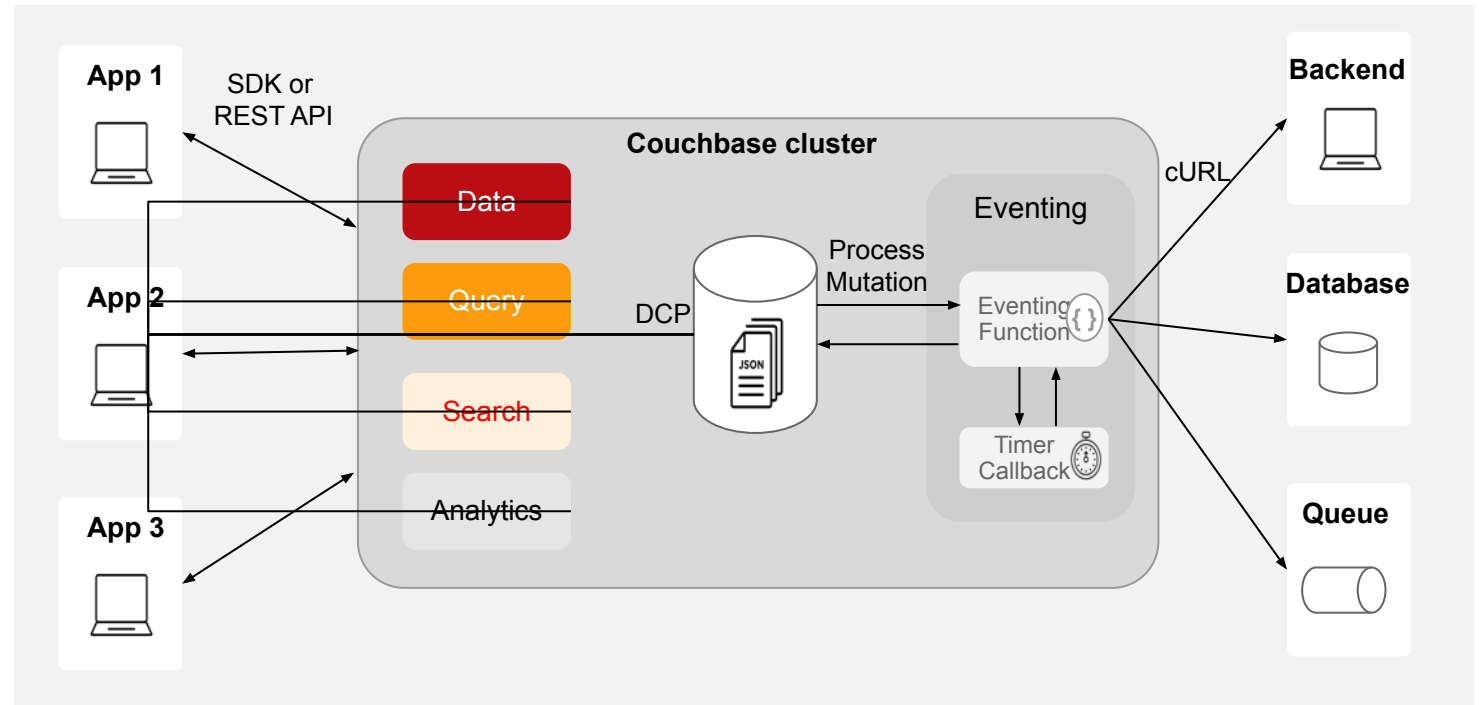
- Event-Condition-Action Model
- JavaScript 기반 : JSON Document의 변경을 분석, 처리에 유리

## • Handler Type : DB Trigger와 유사

- onUpdate
- onDelete

## • 주요 Use case

- 실시간 Document Enrichment
- Threshold 기반 Altering
- Streaming 처리
- 데이터 변경에 대한 확산 작업
- 데이터 Cleansing



## • 비즈니스 로직을 중앙 집중적으로 관리

- Document 데이터 변경에 실시간으로 Trigger되는 Business Logic 구현
- Java Script 기반으로 Cluster 내 Key-Value operation, SQL++ Query 지원
- cURL function을 통해 외부 REST API 호출 가능

# Analytic Service > HTAP(Hybrid Transactional Analytic Processing)

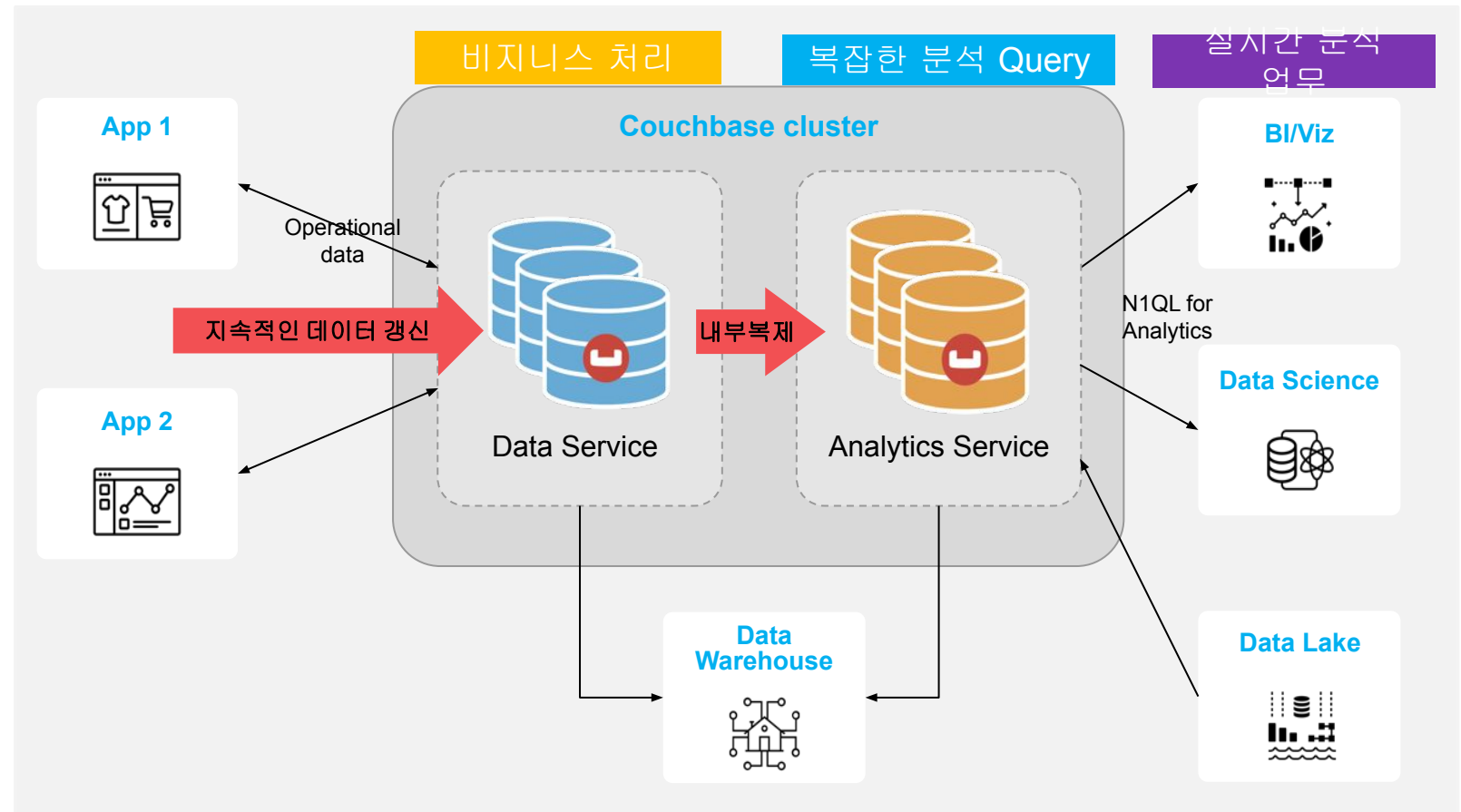
- 데이터를 별도의 시스템으로 추출-변환-로드할 필요 없이 거의 실시간으로 JSON 데이터를 분석
- 운영 데이터나 쿼리 속도를 늦추지 않고 Analytics용 SQL++을 사용하여 Analytics 데이터를 쿼리

## Shadow Copy

- Data Service에 처리되는 데이터를 그대로 Analytics Service로 복제
- 메모리 기반 DCP 사용
- 별도의 CDC/ETL 솔루션 불필요

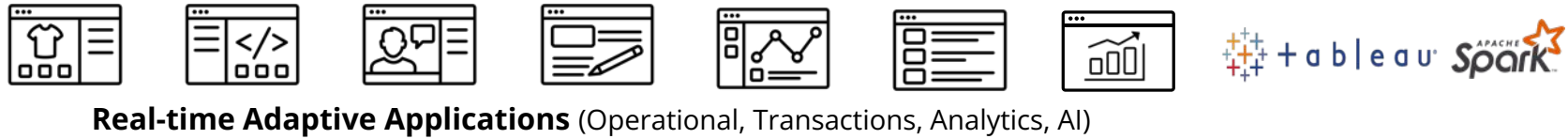
## Parallel Processing

- SQL을 Analytics Service를 구성하는 노드에 분산하여 병렬 처리
- 복잡한 Query 및 ad hoc Query 수행에 적합
- 대용량 데이터 처리
- Tableau Native Connector 제공
- Power BI Native Connector 제공



# Capella Columnar Service : Cloud Data Lake

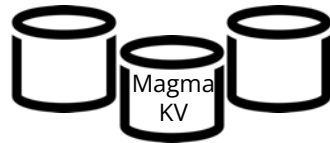
- 내부 JSON 데이터를 비롯하여 다양한 외부 데이터 소스를 통합하여 분석을 수행하는 컬럼 기반 Data Lake 기능 출시



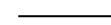
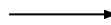
Couchbase  
**CAPELLA**

## Operational Services

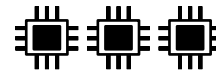
Key-Value, ANSI SQL, Full Text Search



←  
JSON



## Columnar Services



**Independent Compute**  
MPP Query Engine



**Independent Storage**  
Distributed, **Columnar**

**Kafka**

**Object Store Connector**



**Other data stores**  
NoSQL, RDBMS,  
Streaming

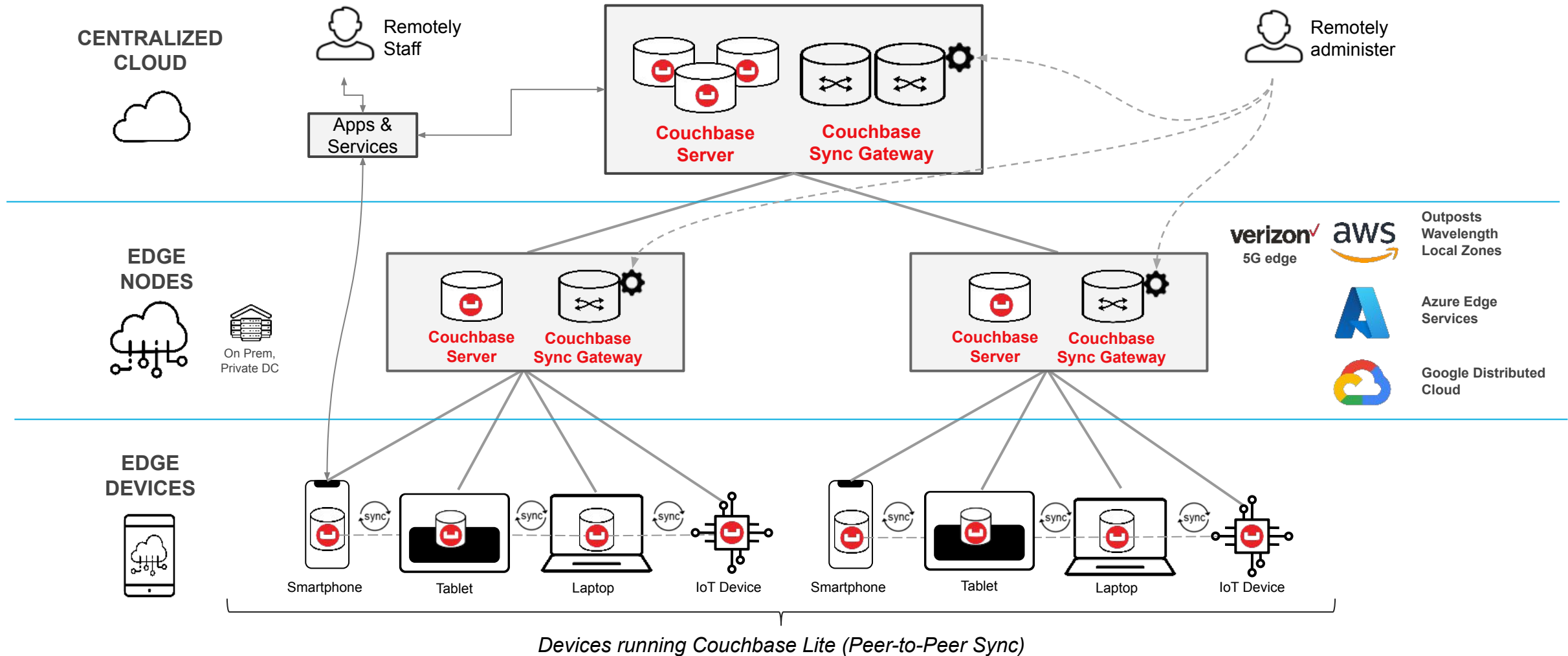


**S3 Files**  
JSON, CSV, TSV,  
Avro, Parquet,...



# Mobile Platform : Mobile/IoT/Edge

- 데이터 관리가 필요한 모든 디바이스에 데이터베이스를 적용할 수 있으며, 데이터 센터의 데이터베이스와 손쉬운 일관성을 유지



# Mobile (Couchbase Lite, Sync Gateway) 지원 범위

- 기능 개요

- JSON Document 모델
- SQL++ Query 지원
- Full Text Search 지원
- Predictive Query 지원

- 지원 개발 플랫폼

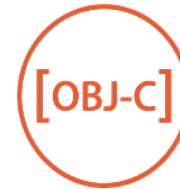
- iOS, macOS
- Kotlin, Java for Android
- C-API

- Sync 방식

- Remote Database with Sync Gateway
- Peer to Peer 동기화 지원
- Star, Mesh 구성 가능



iOS and macOS



Kotlin for  
Android



Java for  
Android



Desktop



Embedded, Desktop  
& Mobile



Desktop & Tablets



iOS & Android

# Deployment

- 완전관리형 클라우드 데이터베이스 서비스(DBaaS)인 **Capella** 출시



## Fully Managed

- 완전관리형 데이터베이스 서비스
- **AWS, GCP, Azure**
- 설치, 구성, 모니터링, 업그레이드 등 모든 운영은 Couchbase가 담당



## Enterprise

- **Bare-metal** 서버, 클라우드 **IaaS** 서버
- **Private Cloud** 서버, **K8S** 컨테이너
- 설치, 구성, 모니터링, 업그레이드 등 모든 운영은 고객이 수행



**Linux, Windows, MacOS, Intel/AMD, ARM**

# Community vs. Enterprise Edition vs. Capella

Source : <https://www.couchbase.com/products/editions/>

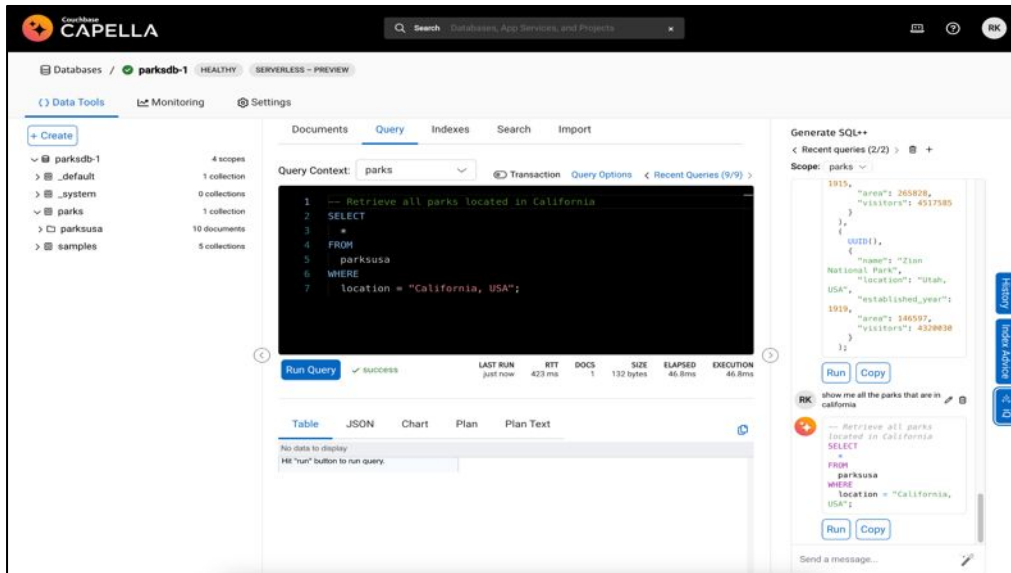
구분	Community Edition	Enterprise Edition	Capella
Cluster 구성	최대 5 노드	제한 없음	제한 없음
	Active-Active Availability	Active-Active Availability	Active-Active Availability
서비스	Data Service	Data Service	Data Service
	Query Service	Query Service	Query Service
	Index Service, No HA	Index Service	Index Service
	Full Text Search Service	Full Text Search Service	Full Text Search Service
	No Analytics Service	Analytics Service	Analytics Service
	No Eventing Service	Eventing Service	Eventing Service
	No Backup Service, import/export	Backup Service	Backup Service
	No XDCR Service (DR)	XDCR Service (DR)	XDCR Service (DR)
	No IQ Service	No IQ Service	IQ Service
	No Columnar Service	No Columnar Service	Columnar Service
운영관리 (성능, 장애, Patch/Upgrade)	By Self	By Self	By Couchbase
	Week Security Features	Strong Security Features	Strong Security Features

# Capella iQ : 생성형 AI 기반 코딩 보조

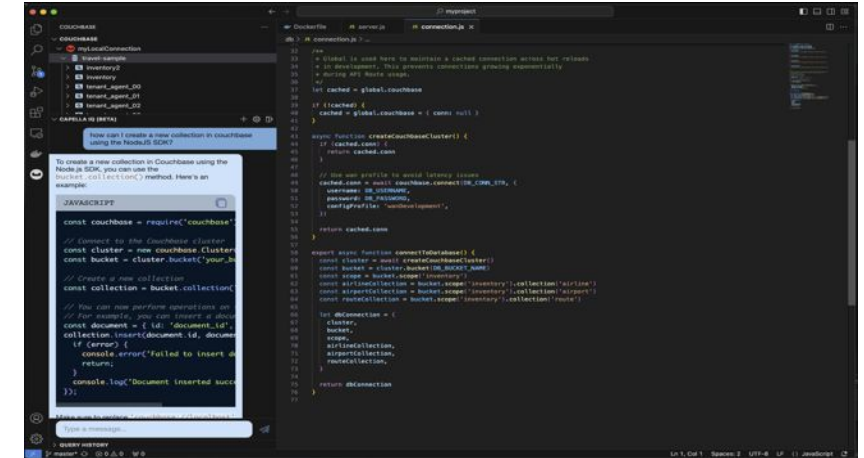
- Generative AI의 LLM을 활용한 Couchbase Capella 전용 Code Assistant
- 자연어로 SQL 및 소스 코딩 지원
- Couchbase 내부 스키마 정보를 활용하여 실제적인 코딩 지원



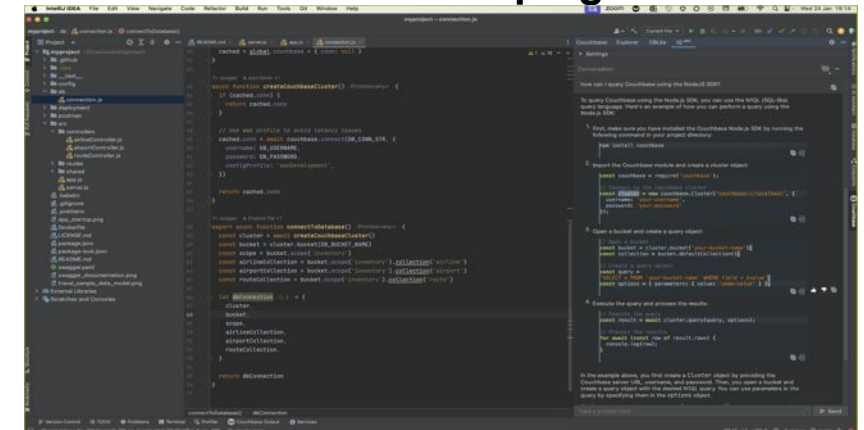
## Capella Workbench



## Visual Studio Code



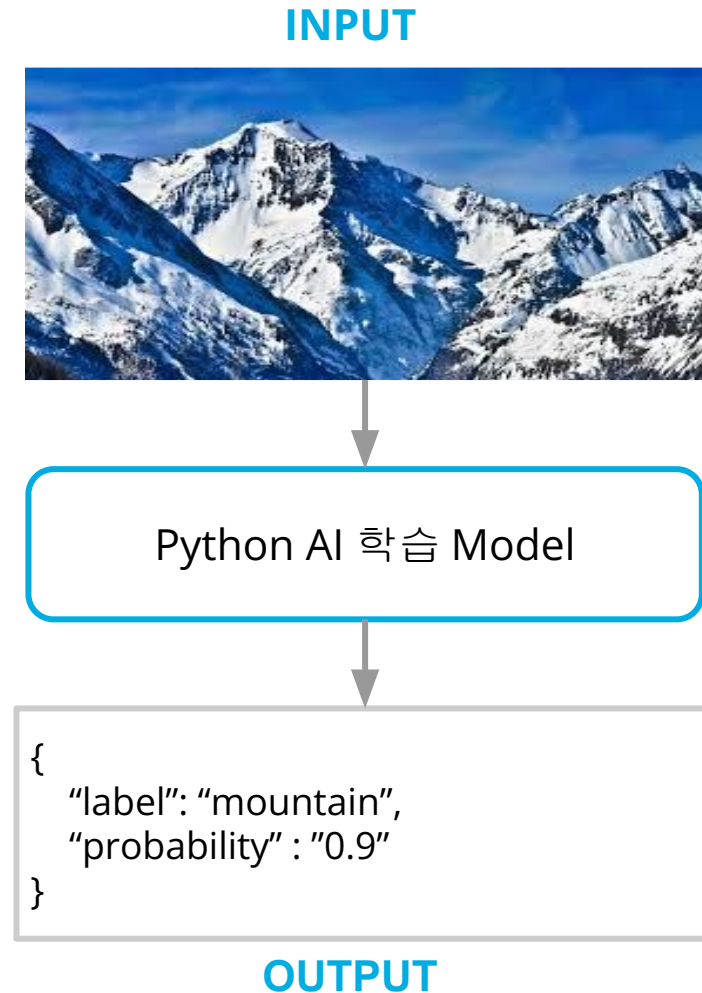
## IntelliJ IDEA plug-in



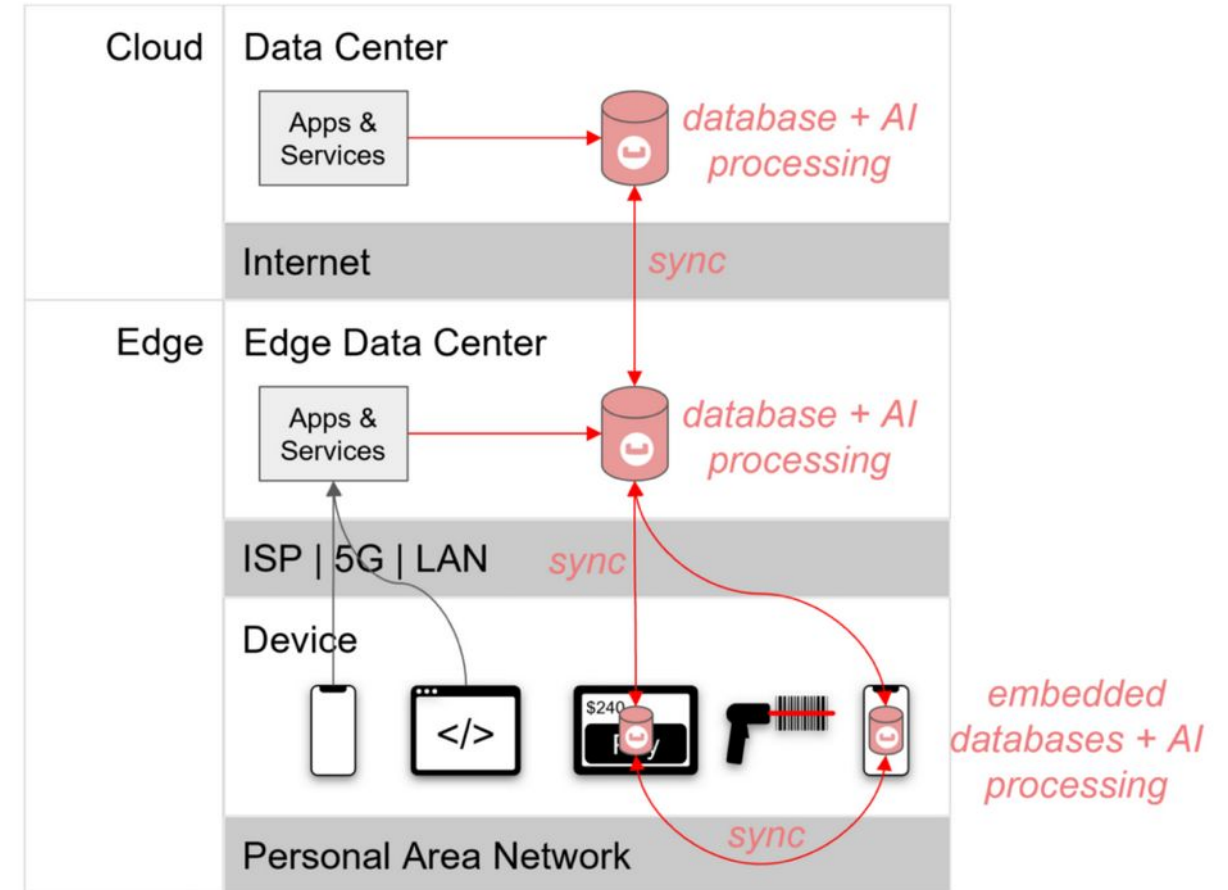


# AI 학습 모델 Embedding & Vector Search

- 대부분의 데이터가 빠르게 생성되고 다양한 의사 결정이 필요한 **Edge** 환경에서 학습 모델을 반영



AI Edge Architecture With Couchbase



# AI-Powered 어플리케이션 지원 데이터 플랫폼

- **AI will drive a transformational impact for businesses**  
AI는 기업에 혁신적인 영향을 미칠 것입니다
- **Applications will be reimagined, evolve existing and build net-new**  
애플리케이션을 재구성하고, 기존 애플리케이션을 발전시키며, 완전히 새로운 애플리케이션을 구축할 것입니다.
- **AI requirements are aligned to Couchbase's sweet spot**  
AI 요구 사항은 Couchbase의 최적 지점에 맞춰져 있습니다.



AI prompts require diverse data	Data complexity and latency is the enemy of AI	AI apps need collocated operational and analytic data	AI data is created and consumed at the edge
JSON is AI's data format	Multimodel NoSQL simplifies architectures	Calculated analytics is an important ingredient	User-facing AI applications are mobile applications



# 개발 지원 문서

Couchbase Documentation

SERVERMOBILECAPELLACLOUD-NATIVE

.NET SDK3.4

C SDK

Go SDK2.7

Java SDK3.5

Kotlin SDK1.2

Node.js SDK4.2

PHP SDK4.1

Python SDK4.1

Ruby SDK3.4

Scala SDK1.5

C++ Transactions

Elasticsearch Connector

Kafka Connector4.2

Spark Connector3.3

Tableau Connector

Power BI Connector

Start

The Couchbase SDK offers a simple, easy-to-use interface for interacting with the Couchbase Server. In this guide, we will walk you through the steps to get started with the Couchbase SDK.

How to use the Couchbase SDK

How to use the Couchbase SDK

How to use the Couchbase SDK

Couchbase Documentation

SERVERMOBILECAPELLACLOUD-NATIVESDKS

SDKS

.NET SDK3.4

C SDK

Go SDK2.7

Java SDK3.5

Kotlin SDK1.2

Node.js SDK4.2

PHP SDK4.1

Getting Started

Start Using the .NET SDK

Data Operations

Query

Search

Sample Application

Transactions

Further Data Ops

Managing Couchbase

Errors & Diagnostics

Learn

References

Project Docs

Start Using the .NET SDK

Data Operations

Query

Search

Sample Application

Transactions

Further Data Ops

Managing Couchbase

Errors & Diagnostics

Learn

References

Project Docs

Data Operations

Data service offers the simplest interface to the Couchbase Server, allowing you to perform CRUD operations on documents. It also offers a simple interface for managing document expiration, and optimistic concurrency control.

Documents

A document refers to an entry in the Couchbase Server, which contains the actual application data. Documents are stored in the Couchbase Server and are managed using the Couchbase Data Platform. Or you can use the Couchbase SDK to interact with the Couchbase Server.

CRUD Operations

The core interface to Couchbase Server is the CRUD interface, which allows you to perform basic operations on documents. The core interface to Couchbase Server is the CRUD interface, which allows you to perform basic operations on documents.

Insert

CSHARP

var document = new {foo = "bar"}; try { var result = await document.InsertAsync(); } catch (DocumentExistsException ex) { Console.WriteLine("Document already exists."); } }

Options may be added to operations using the InsertOptions class.

Insert (with options)

CSHARP

var document = new {foo = "bar"}; var options = new InsertOptions { Upsert = true }; var result = await document.InsertAsync(options); }

Couchbase Documentation

SERVERMOBILECAPELLACLOUD-NATIVESDKS

SDKS

.NET SDK3.4

C SDK

Go SDK2.7

Java SDK3.5

Kotlin SDK1.2

Node.js SDK4.2

PHP SDK4.1

Getting Started

Start Using the .NET SDK

Data Operations

Query

Search

Sample Application

Transactions

Further Data Ops

Managing Couchbase

Errors & Diagnostics

Learn

References

Project Docs

Start Using the .NET SDK

Data Operations

Query

Search

Sample Application

Transactions

Further Data Ops

Managing Couchbase

Errors & Diagnostics

Learn

References

Project Docs

Sample Application

Discover how to program interactions with the Couchbase Server via the data, query, and search services — using the Travel Sample Application with the built-in Travel Sample data Bucket. Discover how to program interactions with Couchbase Server 7.X via the Data, Query, and Search services — using the Travel Sample Application with the built-in Travel Sample data Bucket.

Quick Start

Fetch the Couchbase .NET SDK travel-sample Application REST Backend from github:

CONSOLE

git clone https://github.com/couchbaselabs/try-cb-dotnet.git  
cd try-cb-dotnet

With Docker installed, you should now be able to run a bare-bones copy of Couchbase Server, load the travel-sample, add indexes, install the sample-application and its frontend, all by running a single command:

CONSOLE

docker-compose --profile local up

Running the code against your own development Couchbase server.

For Couchbase Server 7.1, make sure that you have at least one node each of data; query; index; and search. For a development box, mixing more than one of these on a single node (given enough memory resources) is perfectly acceptable.

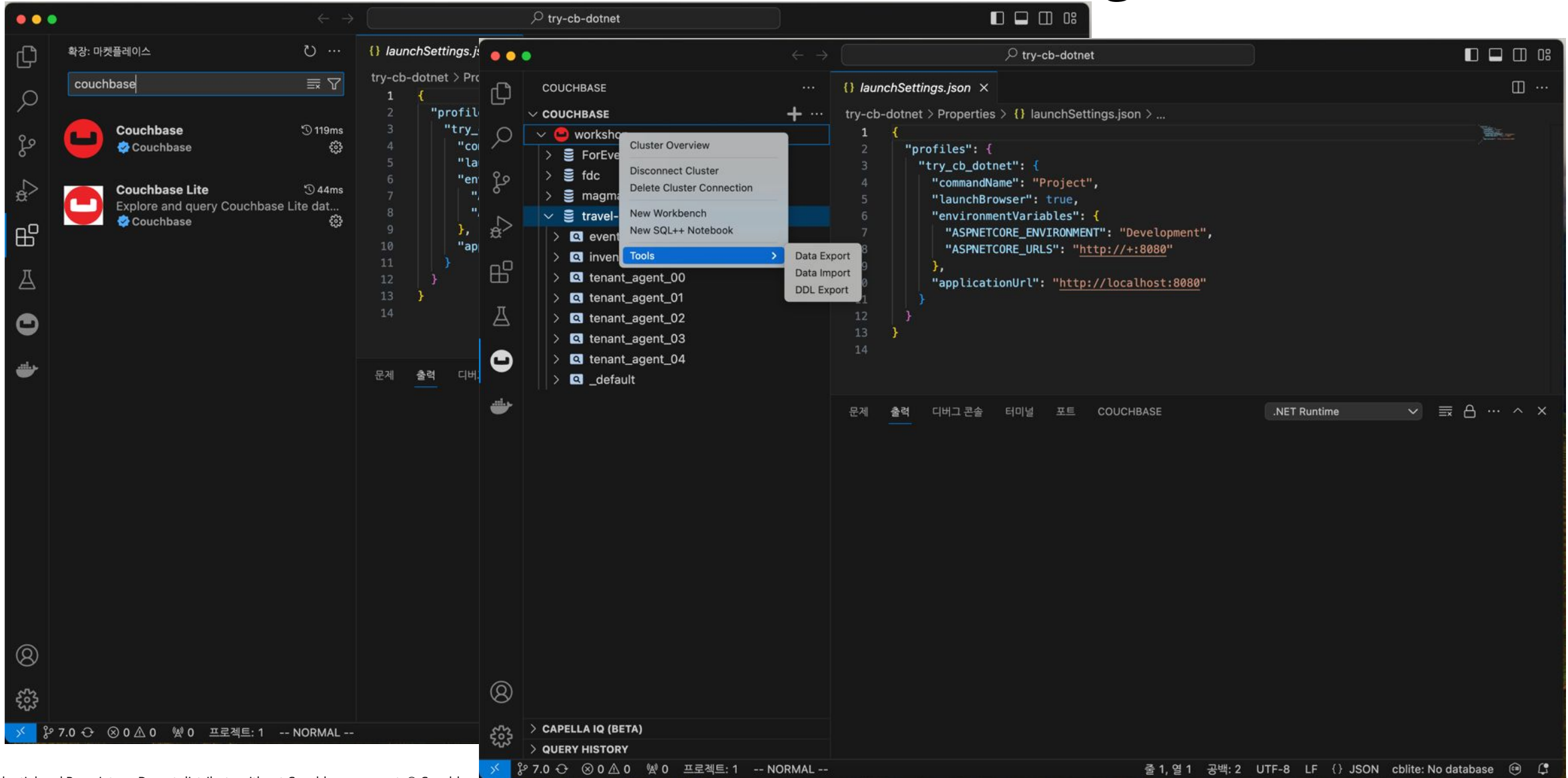
If you have yet to install Couchbase Server in your development environment start here.

Then load up the Travel Sample Bucket, using either the Web interface or the command line. You will also need to create a Search Index — Query indexes are taken care of by the Sample Bucket.

Confidential and Proprietary. Do not distribute without Couchbase consent. © Couchbase 2025. All rights reserved.

41

# 개발 지원 툴(Visual Studio Code, Couchbase Plug-in)



# 개발 지원 툴(VS Studio, Couchbase Plug-in)

The screenshot shows the VS Studio interface with the Couchbase Plug-in. The left sidebar displays the project structure under 'COUCHBASE', including 'workshop', 'ForEvent', 'fdc', 'magma', 'travel-sample', 'eventtest', and 'inventory'. The 'inventory' folder is expanded, showing 'airport (2.0k)', 'airline (187)', 'route (24.0k)', 'landmark (4.5k)', 'hotel (917)', and several 'tenant\_agent' files. The main editor shows a SQL query in 'workbench-1.sqlpp':

```
1 select address, city, country, geo.lat, geo.lon from `travel-sample`.inventory.hotel limit 5;
```

Below the query, the Couchbase status bar shows: **success** | RTT: 302 MS | SIZE: 526 Bytes | ELAPSED: 2 MS | EXECUTION: 2 MS | DOCS: 5 docs. The 'Table' tab is selected, displaying the following data:

address	city	country	lat	lon
Capstone Road, ME7 3JE	Medway	United Kingdom	51.35785	0.55818
57-59 Balmoral Road, ME7 4NT	Gillingham	United Kingdom	51.38624	0.55078
6 rue aux Juifs	Giverny	France	49.078069	1.520866
5 rue de la Dîme	Giverny	France	49.0763077	1.5234464
99 Gordon Street	Glasgow	United Kingdom	55.86458	-4.26655

An inset window shows the 'Plan' tab of the query execution plan. It displays a sequence of steps: **Authorize** (00:00.000 (1.3%)), **PrimaryScan3** (00:00.000 (40%)), **Fetch** (00:00.000 (37.6%)), **Project** (00:00.000 (17.7%)), **Limit** (00:00.000 (0.2%)), and **Stream** (00:00.000 (2.5%)).

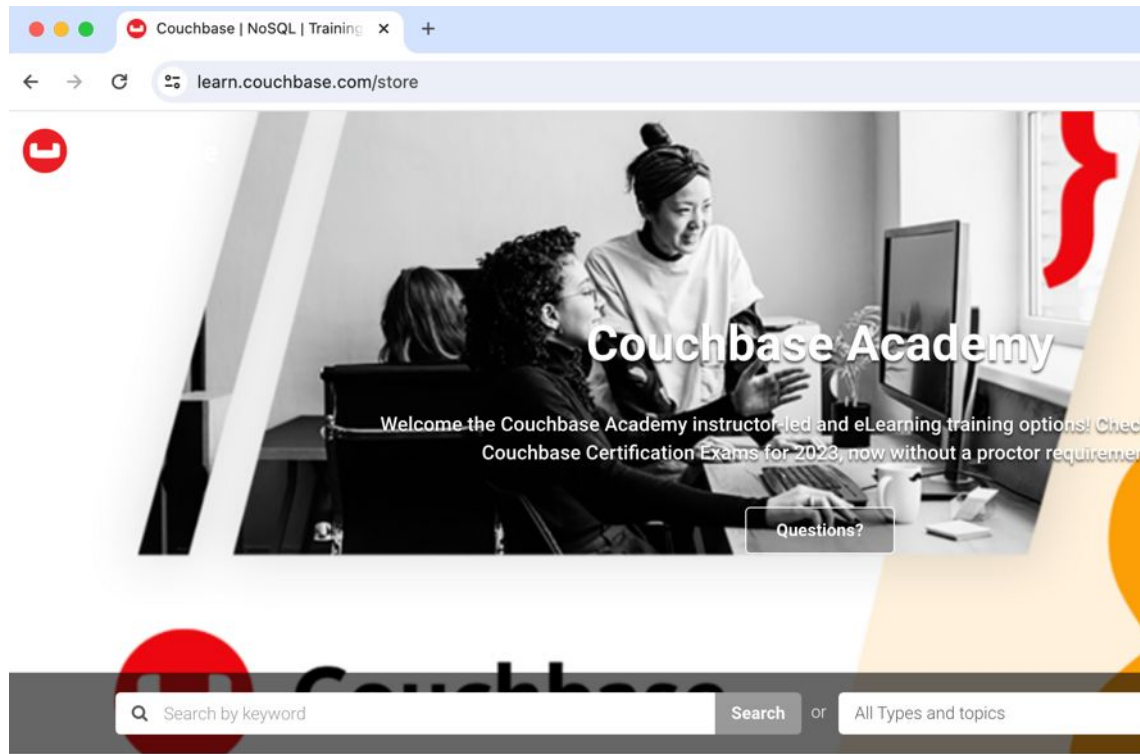
# Capella Webinar

구분	일자	Webinar 주제
시리즈 1	2023-11-29	SQL ++ & Index
시리즈 2	2023-12-20	Full Text Search & UDF
시리즈 3	2024-01-24	Eventing & Analytics
<b>시리즈 4</b>	<b>2024-03-27</b>	<b>Couchbase Lite &amp; Sync Gateway</b>
시리즈 5	2024-04-24	SDK & Development
시리즈 6	2024-06-26	Capella New Features



# Couchbase Academy

<https://learn.couchbase.com/store>



## Upcoming Sessions

February	CD410: Advanced N1QL Course: Tuning and Optimization - APAC Virtual (GMT+8)		
20	Starting: 02/20/2024 @ 09:00 AM (GMT+08:00) Singapore	Ending: 02/23/2024 @ 05:00 PM (GMT+08:00) Singapore	Type: Multi-day Session

<https://e-learning-labs.s3.us-west-2.amazonaws.com/cb130cs-lab-instructions-capella/index.html>

**CB140a: Couchbase Associate Android Developer With Capella Course**

This course showcases and demonstrates how to create a new Android application using Couchbase Mobile and Couchbase Capella, our fully managed DBaaS service. The following 7 modules provide fundamental instruction on building an Android application with or without a pre-existing database. To that end, we walk through the essential functionality of Couchbase Capella, the benefits of a fully managed NoSQL database, and how that database interacts with Couchbase Mobile. [Read More](#)

Rating  
★★★★★

Difficulty  
■ ■ ■

Content  
8 modules

Price  
Free

Add

**CB130cs: Couchbase Associate .NET Developer With Capella Course**

This course leverages the full power of Couchbase Capella, our cloud offering that allows Developers access to a Couchbase database through our DBaaS service and Couchbase 7. The following 8 courses provide a fundamental understanding of the Couchbase NoSQL database and essential functionality. Throughout these courses, we share the basics of SQL vs. NoSQL, obtaining and using the Couchbase Capella UI, modeling data to the benefit of Couchbase and an example application. [Read More](#)

Rating  
★★★★★

Difficulty  
■ ■ ■

Content  
9 modules

Price  
Free

Add

**CB130n: Couchbase Associate Node.js Developer With Capella Course**

This newly revamped course shows how to leverage the full power of Couchbase 7 as a service with Couchbase Capella. The following 8 courses provide a fundamental understanding of the Couchbase NoSQL database and essential functionality. Throughout these courses, we share the basics of SQL vs. NoSQL, how to sign up for Couchbase Capella, modeling data to the benefit of Couchbase, and an example application you will build. Learners will also walk through the basics of Couchbase. [Read More](#)

Rating  
★★★★★

Difficulty  
■ ■ ■

Content  
9 modules

Price  
Free

Add

**CB130j: Couchbase Associate Java Developer With Capella Course**

This newly revamped course leverages the full power of Couchbase 7 and supports Couchbase Capella. The following 8 courses provide a fundamental understanding of the Couchbase NoSQL database and essential functionality. Throughout these courses, we share the basics of SQL vs. NoSQL, obtaining and downloading Couchbase, modeling data to the benefit of Couchbase and an example application you will build. Learners will also walk through the basics of Couchbase's N1QL. [Read More](#)

Rating  
★★★★★

Difficulty  
■ ■ ■

Content  
9 modules

Price  
Free

Add

# 결론적으로 > Modern AI-Powered 어플리케이션을 위한 데이터 플랫폼

## JSON Document 직관적이고

- 사람이 인지 하는 세상을 그대로 데이터 모델로 사용
- 복잡한 정규화 과정없이 직관적인 방식으로 어플리케이션 개발/운영

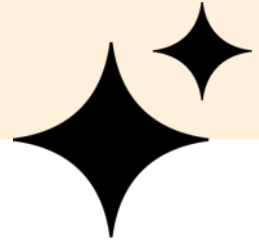
## SQL, Generative AI 익숙하고, 쉽게

- NoSQL 이지만 표준 SQL을 지원
- 생성적 AI인 Capella IQ 지원으로 더 손쉬운 개발이 가능

## Data Platform 일관적 적용

- Key/Value 데이터서비스에서 분석서비스, 모바일 앱서비스까지 일관성있게 업무에 적용 가능
- 센서, 모바일, 퍼스널컴퓨터, 데이터센터 서버, 쿠버네티스, 클라우드에 동일한 데이터플랫폼 적용으로 개발의 일관성 뿐만 아니라 데이터의 일관성도 보장

**Enterprise 에서 요구하는 성능, 안정성, 보안성**



# Thank you!



[paul.son@couchbase.com](mailto:paul.son@couchbase.com)

[www.couchbase.com](http://www.couchbase.com)

[cloud.couchbase.com](https://cloud.couchbase.com)



**Couchbase**

