Couchbase

Couchbase Capella Workshop
> GenAI (LLM, RAG)

2시에 시작하겠습니다.

2024.10.30

손 광락, Solutions Engineer

Capella Webinar

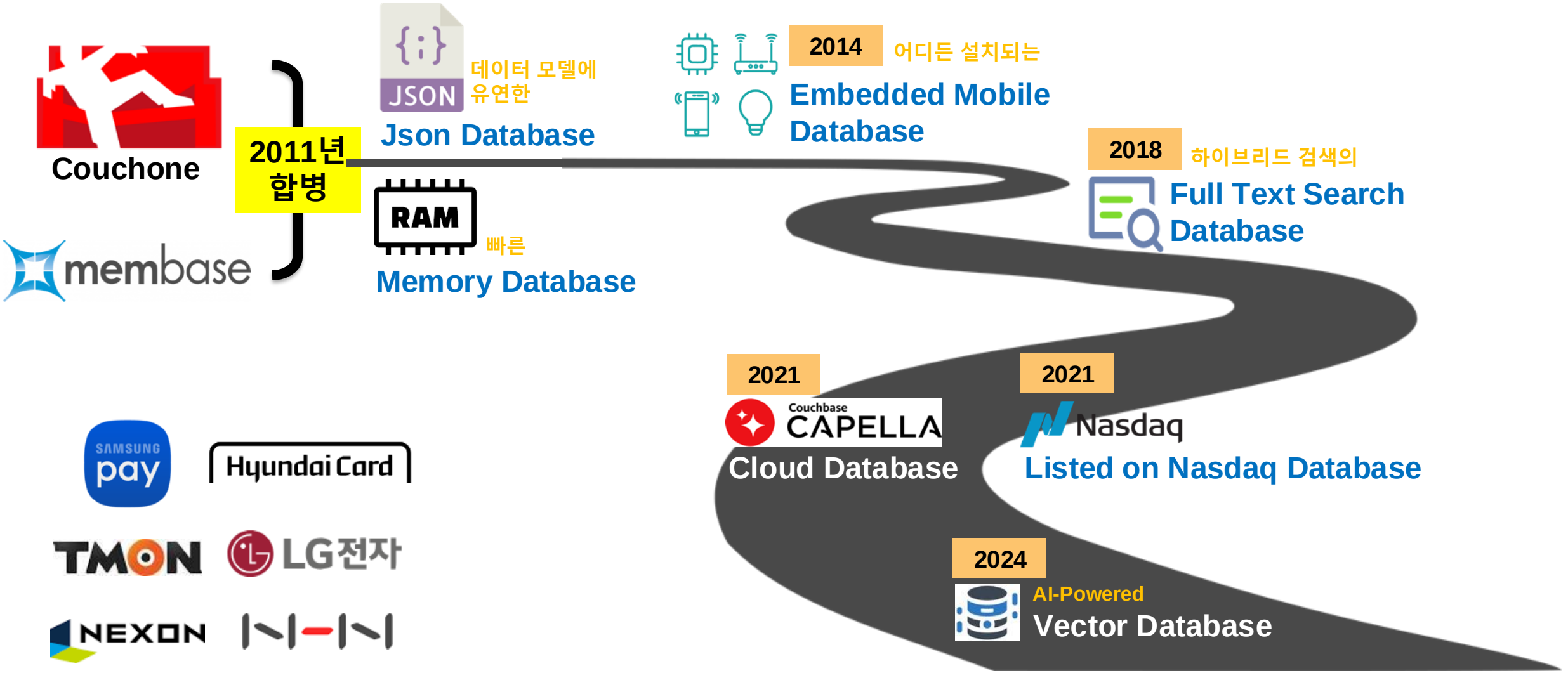
구분	일자	Webinar 주제
시리즈 1	2024-09-25	벡터 검색을 활용한 AI Powered 어플리케이션 구축
시리즈 2	**2023-10-30**	**벡터 검색을 활용한 GenAI(LLM/RAG) 어플리케이션 구축**
시리즈 3	2024-11-27	벡터 검색을 활용한 Mobile On-Device AI 어플리케이션 구축

Agenda

- 14:00 – 14:10 등록 확인 & 실습 환경 확인
- 14:10 – 14:35 카우치베이스 소개
- 14:35 – 15:00 AI, Vector 와 Vector 검색, GenAI
- 15:00 – 15:10 Break
- 15:10 – 15:25 Capella/Couchbase 사용법 실습
- 15:25 – 15:50 GenAI(LLM,RAG) 실습/데모
- 15:50 – 16:00 Q & A

Couchbase
Introduction & Architecture

1 개요 : Couchbase 회사 소개



1 개요 : Couchbase 도입 사례

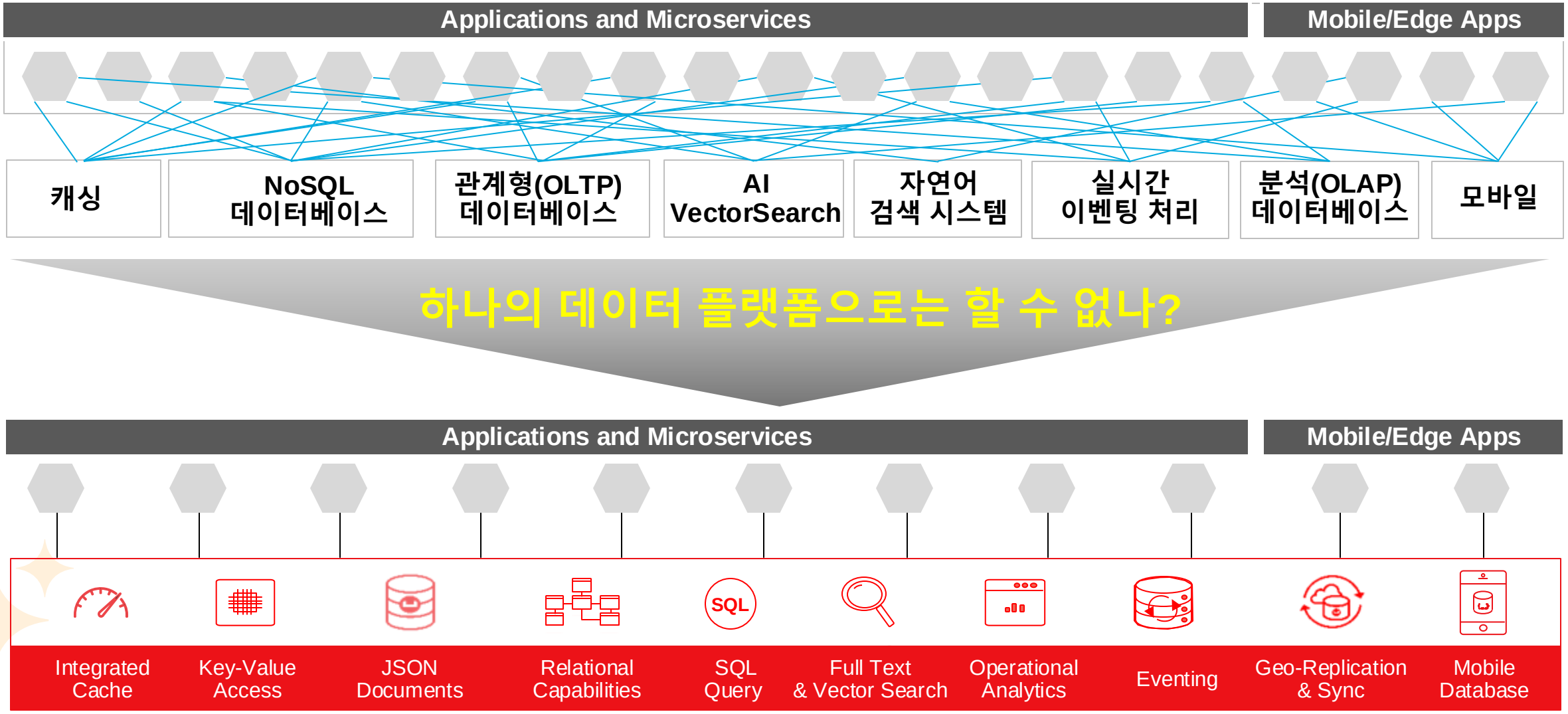
	LinkedIn	TESCO	amadeus	COMCAST	UNITED
Application	캐싱 & 싱글뷰를 위한 세션 스토어	리얼 타임 프라이싱, 제품 캐탈로그, 재고관리	비행편 가용성, 예약, 가격분석등	Customer 360 싱글뷰, 'Unified notes'App지원	리얼타임 승무원 분석, 일정 및 리소스 관리
Performance	**2백만+** 읽기/초당	**1천만+** SKUs	**8백만** Ops / 초당	**2.1억개** 다큐먼트	**4.1만** 종업원
	1천만 쿼리/초당	**3만5천** 요청/초당.	**<2.5ms** 반응시간	**10만** 사용자	**1.5억+** 이용객

Walmart, cars.com, Fanatics, Office Depot OfficeMax, Staples, Tesco, Best Buy, Marriott, amadeus, Sabre, Orbitz, tripadvisor, Hilton, FICO, Revolut, PayPal, Wells Fargo, VISA, Pfizer, everyday health, BD, Cochlear, Monsenso, Comcast, KDDI, at&t, Viber, Telefónica, sky, Thomson Reuters, Fandango, Gannett, Disney, nielsen, EA, betfair, zynga, Blizzard, Playtika, SGN, verizon, Sierra Wireless, Chamberlain, GE

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

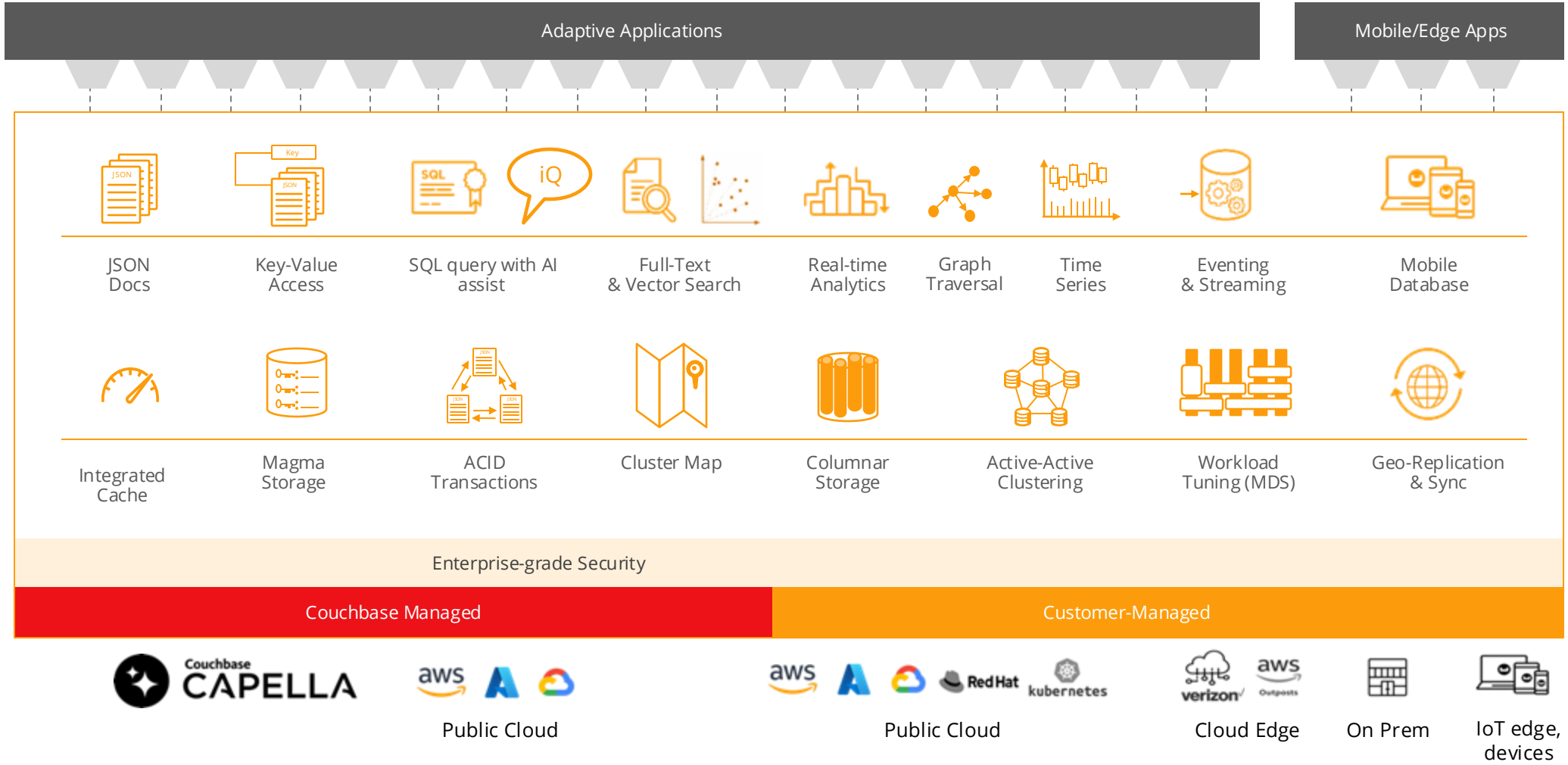
6 of the Top 10 Broadcast Companies
6 of the Top 10 Online Casino Gaming Companies
2 of the Top 2 IoT Platforms

1 개요 : 하나의 App을 만들기 위해 다양한 데이터 관리 솔루션이 필요.

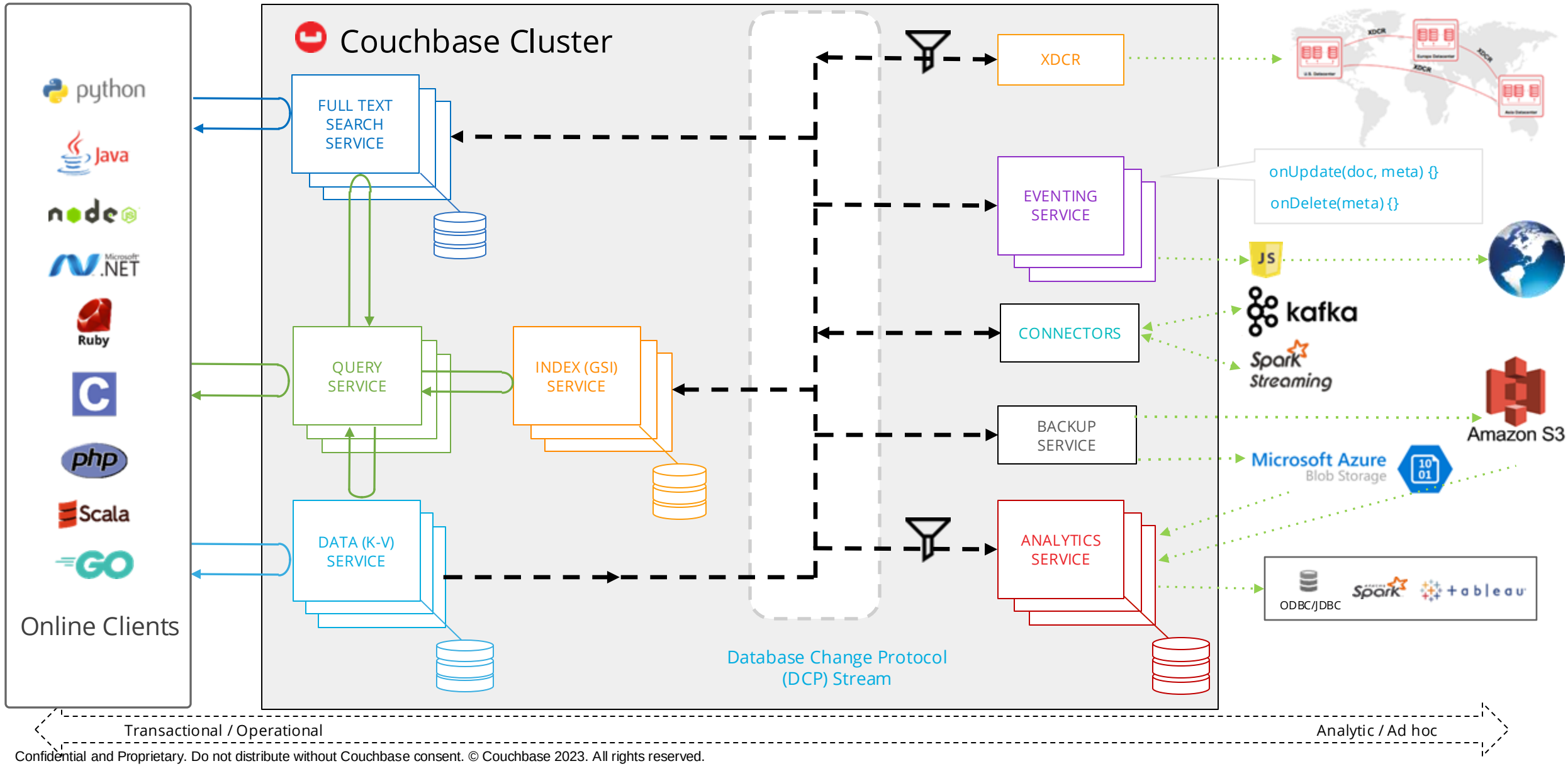


1 개요 : Enterprise 데이터 플랫폼

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.



2 아키텍처 : 메모리 기반 Micro Service 아키텍처

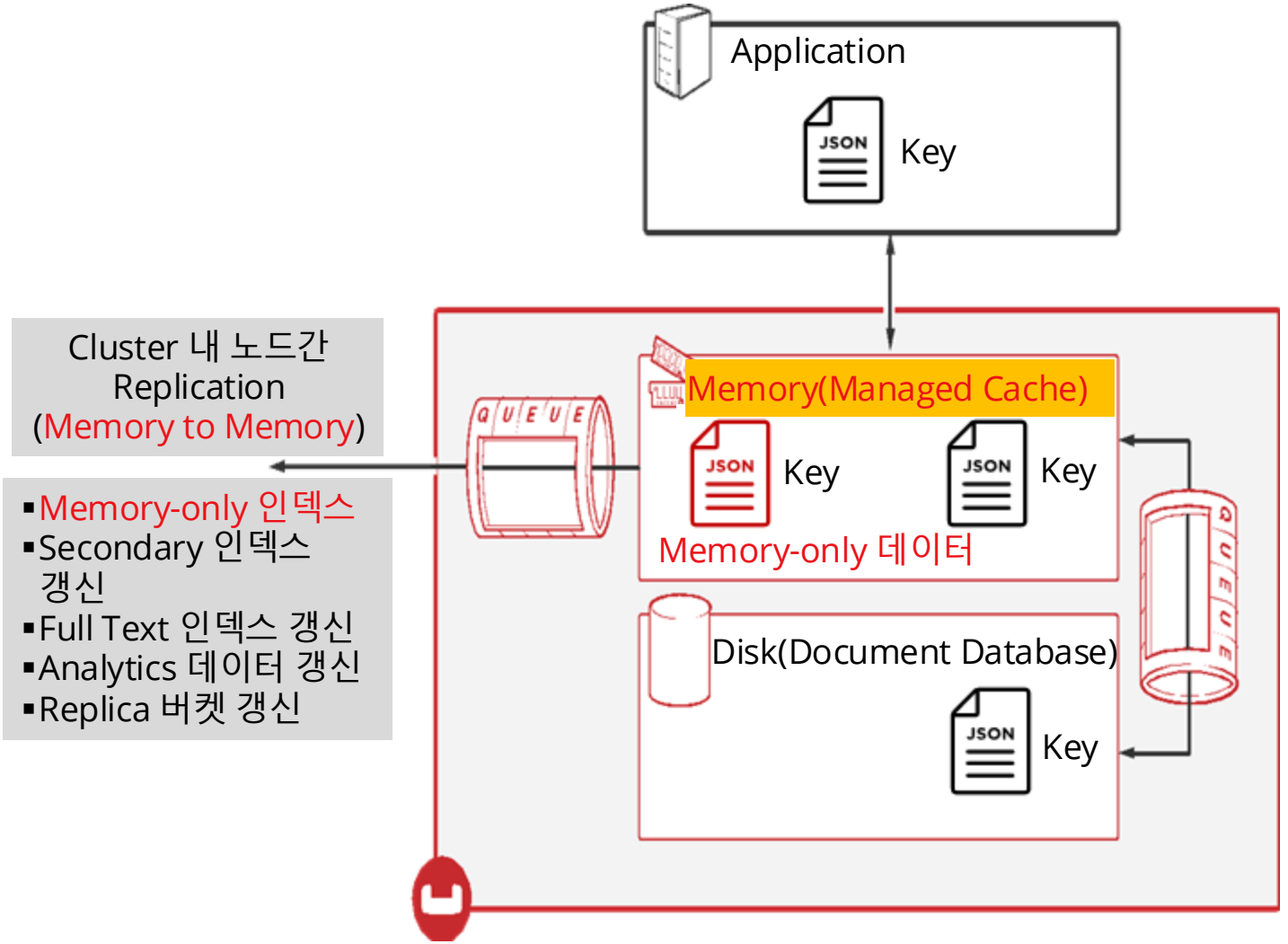


2 아키텍처 : Memory First 아키텍처

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

- **인메모리 Key-Value 오퍼레이션**
  - 특정 Key를 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
  - Value가 단순 수치나 배열이 아닌 JSON 도큐먼트 자체
  - JSON 도큐먼트 처리가 메모리 우선 방식

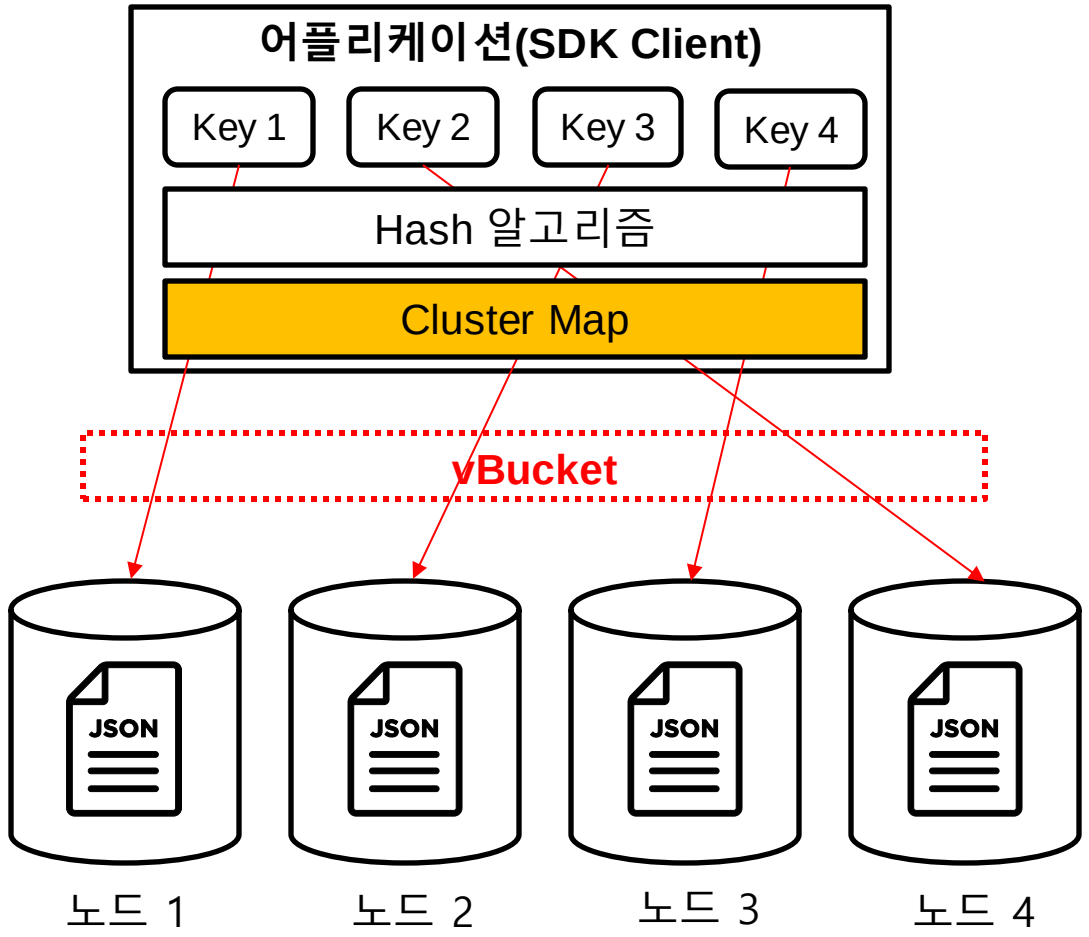


2 아키텍처 : 분산 병렬, Master Node-less 아키텍처

데이터를 다수의 노드에 Key 기반 자동 분산 저장하며, 별도의 마스트 노드없이 모든 노드에서 병렬 처리를 수행함.

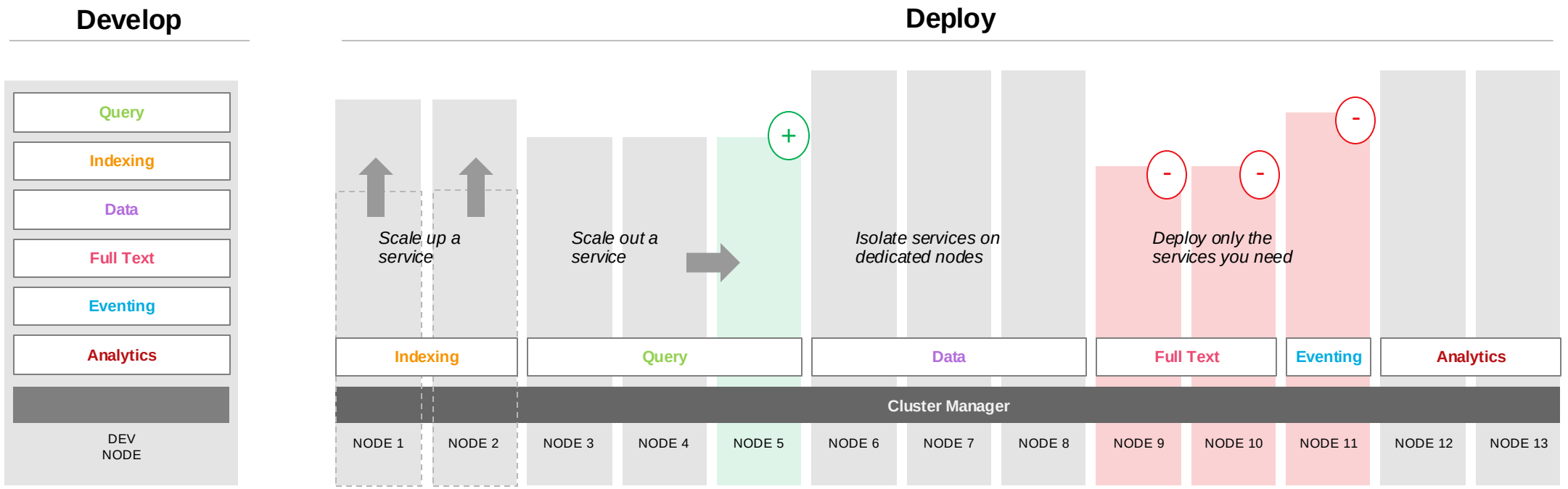
- **Key 기반 자동 분산 아키텍처**
  - 대량의 Key-Value 처리를 노드 별로 분산하여 성능 향상
  - 최대한 균등하게 분산 저장 가능, 특정 노드에 편중되는 현상(Data Skew) 방지
  - 별도의 분산 정책 불필요
- **Couchbase**
  - Key에 대한 Hash 알고리즘 적용으로 자동 분산
  - 노드 추가 시, Rebalancing을 통해 Key 재 분산 수행

- **Master Node-less 아키텍처**
  - 어플리케이션의 Key-Value Operation 시, 해당 Key에 매핑된 특정 노드에 직접 접근
  - 모든 노드가 어플리케이션 측면에서 Active 노드 역할
- **Couchbase**
  - 어플리케이션이 데이터 처리를 구현하기 위해 SDK 활용
  - 데이터 분산 정보(Cluster Map)를 지속적으로 SDK Client에 Update



2 아키텍처 : 자원 절약형 다차원 독립 확장

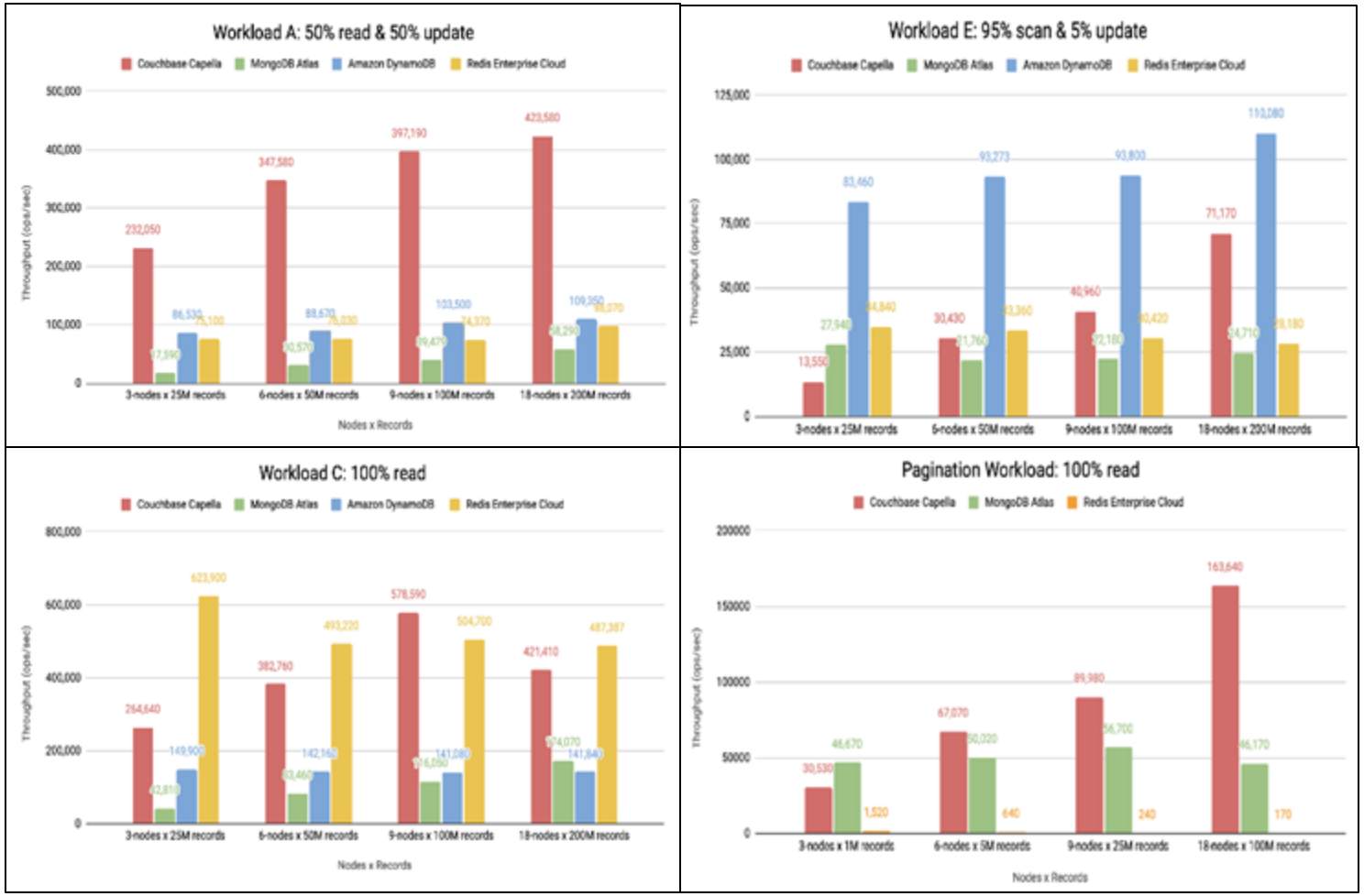
Couchbase는 Multi-Dimensional Scaling 기능으로 서비스 별 Workload 분산 및 독립성을 보장합니다.



- **Service 단위 하드웨어 자원 최적화**
  - 각 Service 별 시스템 자원을 독립적으로 할당
  - **각 Service에서 수행되는 작업이 다른 Service에 영향을 최소화**, 예를 들어 Analytics Service에서 복잡한 작업을 수행하여 시스템 자원을 많이 사용해도 Data Service 혹은 Query Service에서 수행하는 Operational 작업에는 영향이 없는 구조

3 성능 : NoSQL Database 벤치마크에서 탁월한 우위

<Yahoo! Cloud Serving Benchmark(YCSB-NoSQL Benchmark)>



출처 : https://www.altoros.com/blog/couchbase-capella-vs-mongodb-atlas-vs-amazon-dynamodb-vs-redis-enterprise-cloud/

Altoros, April 2023

MongoDB

- Struggles to scale, strongest at 3 nodes
- Price performance makes it worse
- Weakest performer

DynamoDB

- Excels at scans, but throws excessive errors
- Challenged across other workloads

Redis

- Excels at read-only, Capella meets at scale
- Regularly used as cache for Atlas or DynamoDB
- “Fails” Pagination workload

Capella

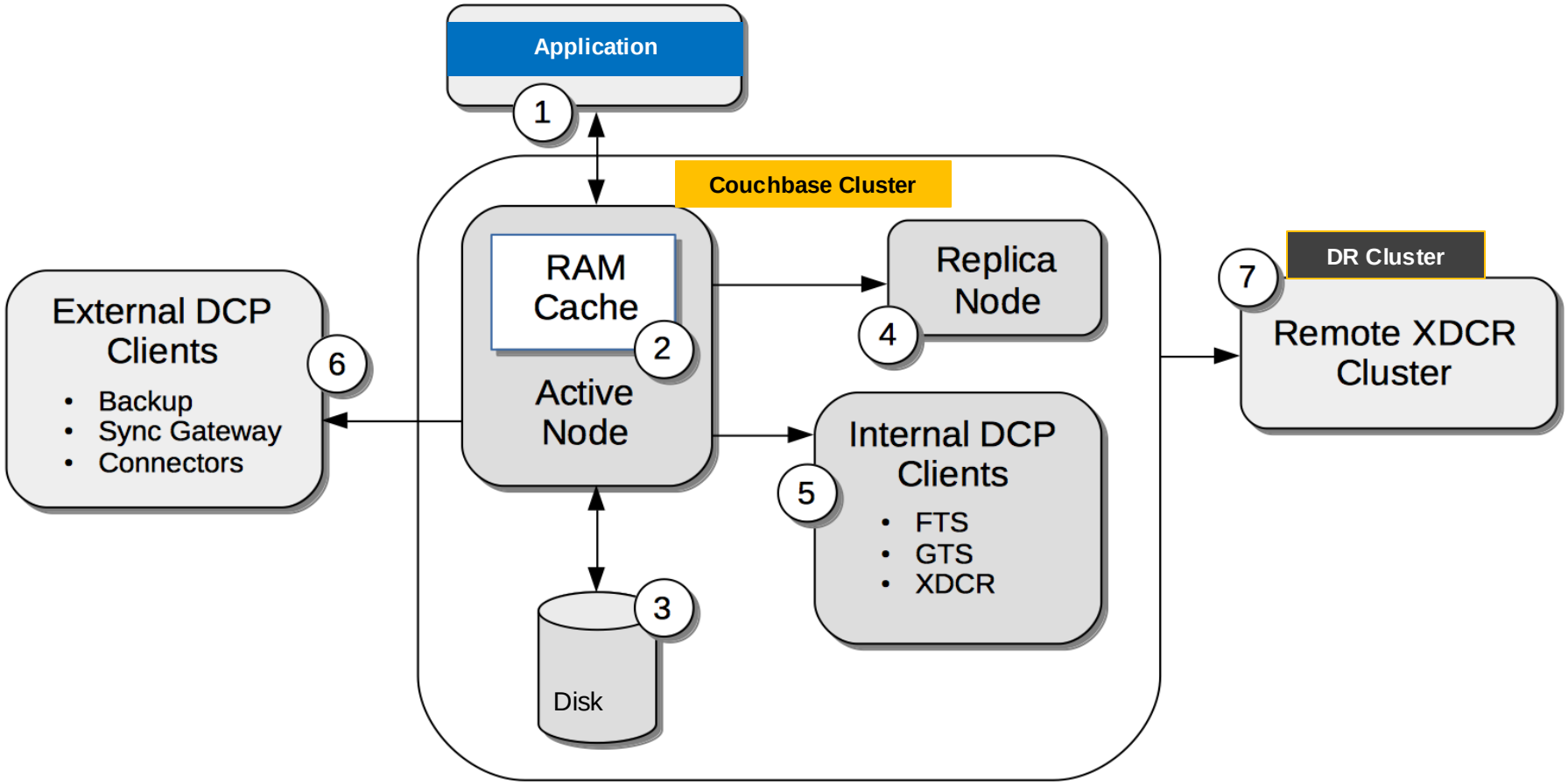
- Excels at Read and Write
- Scales effectively for multiple workloads
- Best price-performance

	Couchbase Capella	MongoDB Atlas	Azure CosmosDB\ (MongoDB API)
	90 /116	84 /116	71 /116
DEPLOYMENT	28 /32	27 /32	21 /32
MANAGEMENT	19 /25	19 /25	19 /25
SUPPORT	11 /16	12 /16	9 /16
PERFORMANCE	20 /26	13 /26	17 /26
PRICING	12 /17	13 /17	5 /17

출처 : https://benchant.com/navigator/dbaas

3 성능 : End-to-End 압축으로 네트워크/스토리지 비용 절감

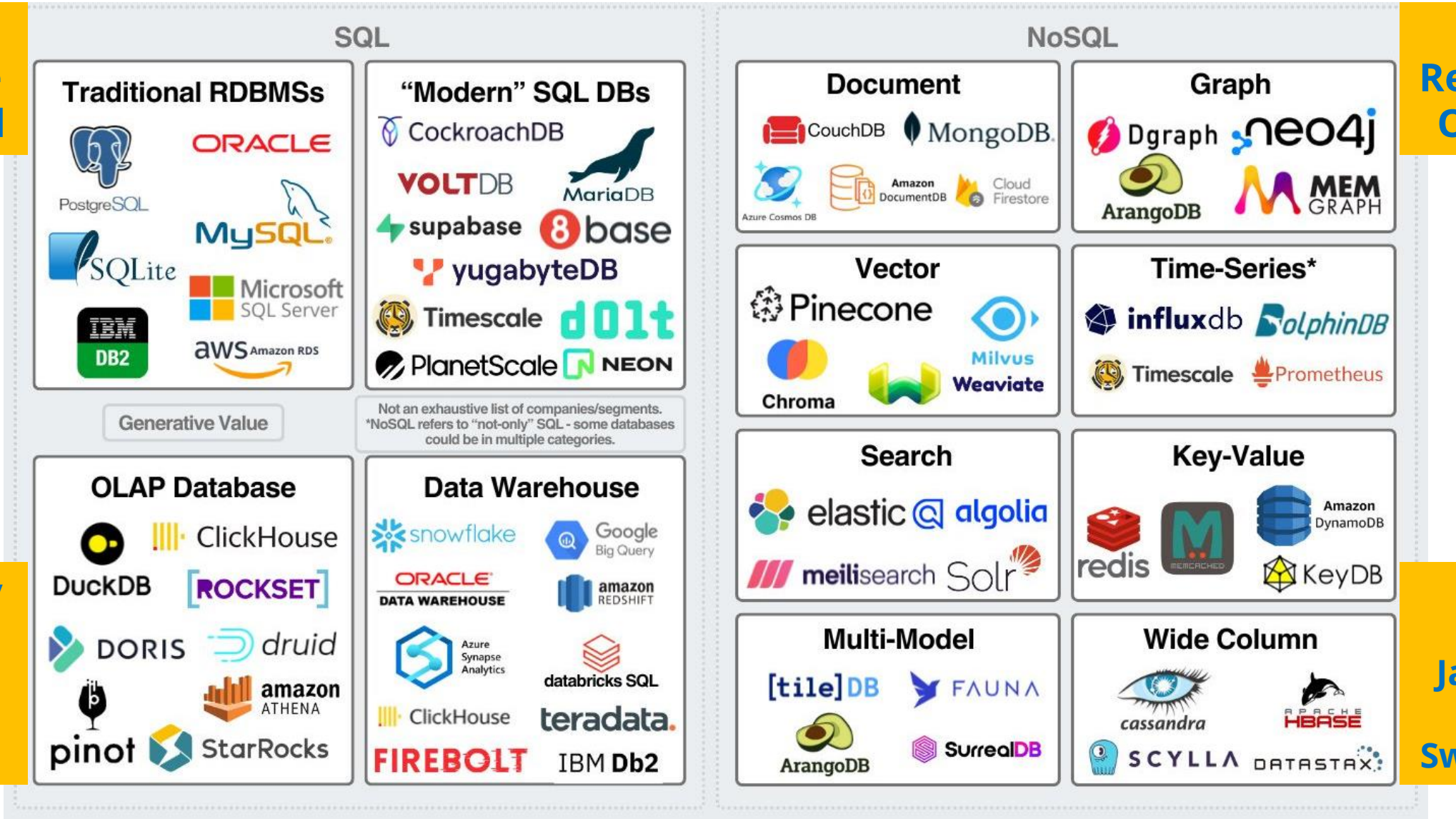
Couchbase는 데이터를 압축하여 전송하며 압축된 형태로 메모리와 디스크에 보관됩니다. 즉, 네트워크 전송량, 메모리 및 디스크 사용량을 줄려 성능 및 운영비용을 절감할 수 있습니다.



4 데이터 모델 : SQL(Table)과 NoSQL(Real World)

1970 Machine Oriented

Assembly Cobol C Pascal Perl



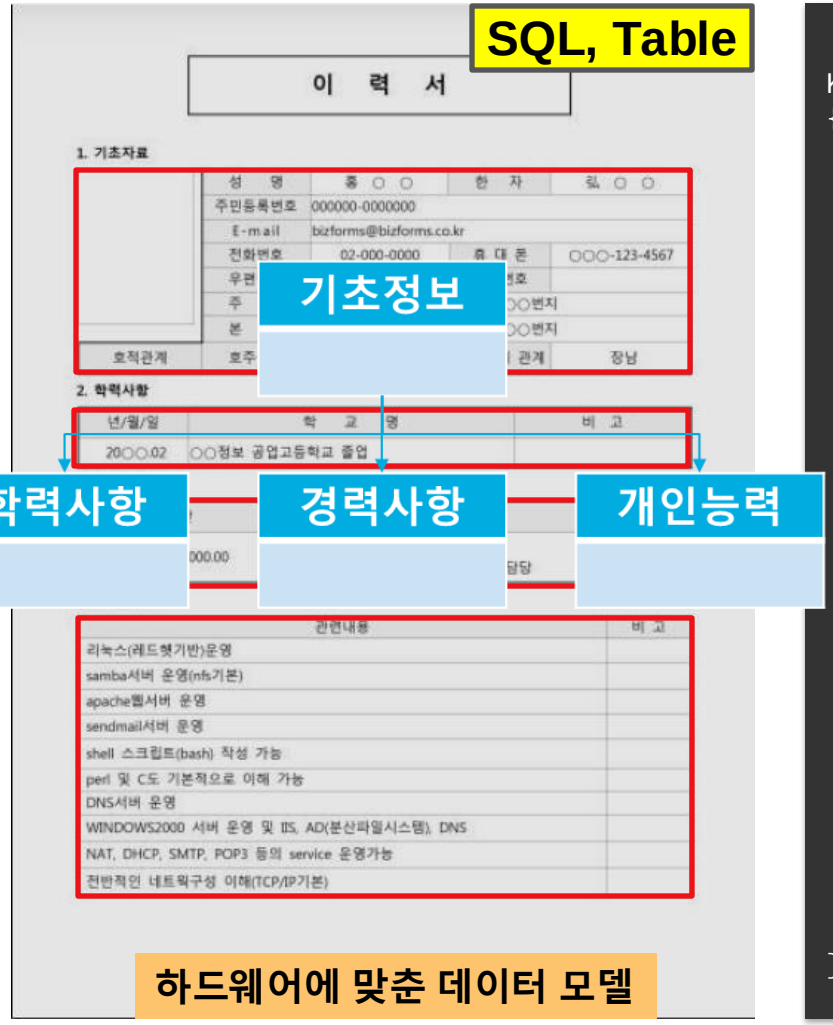
2020 Real World Oriented

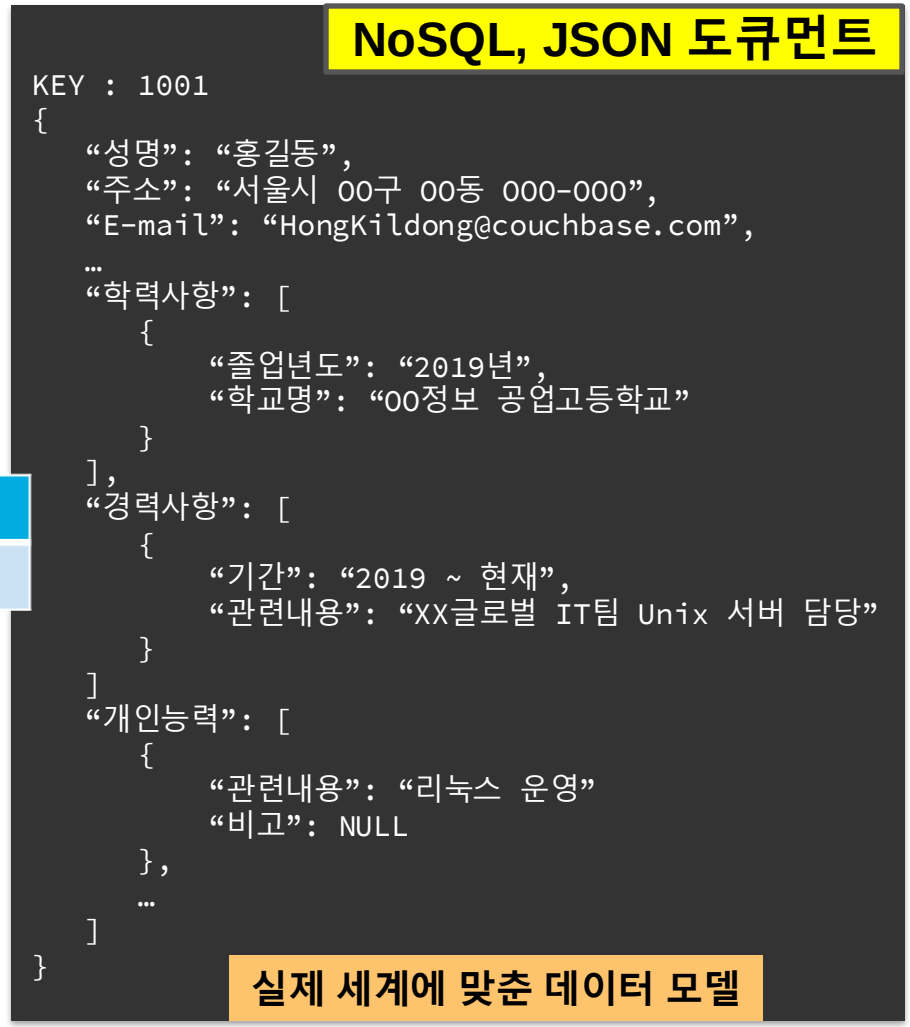
JAVA C# JavaScript Python Swift/Kotlin

4 데이터 모델 : JSON 도큐먼트

JSON은 텍스트로 이루어져 있으므로, 사람과 기계 모두 읽고 쓰기 쉽다. 프로그래밍 언어와 플랫폼에 독립적이므로, 서로 다른 시스템 간에 객체를 교환하기에 좋다.

- 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





4 데이터 모델 : ANSI 표준 SQL vs. 자체 Query 언어

2018년 Q4 고객에 대해서 각 미팅에 소요된 시간, 해당 고객에 모든 미팅에 소요된 시간 대비 해당 미팅의 소요 시간 비율, 순위 등을 조회하는 Query에 대해 Couchbase와 타 NoSQL 솔루션 비교입니다.

```
SELECT
  c.name,
  a.title,
  a.actduration,
  a.startDate,
  SUM(a.actduration) OVER ( PARTITION BY
    c.name ORDER BYc.name, a.startDate )running_total,
  TRUNC(100*(a.actduration/SUM(a.actduration)
      OVER(PARTITION BYc.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:.....
```

- **Couchbase**
  - ANSI 기반 SQL
  - Join, Windows Function 등 지원

4 데이터 모델 : ANSI 표준 SQL++

- 개요
  - **ANSI 2003 SQL 기반**
  - Multi-core 병렬 수행에 최적화된 Query Engine
  - **Cost-based Optimizer 지원**
- 확장 SQL
  - NEST/UNNEST : Embedded Object, Arrays 지원
  - IS EMPTY/IS MISSING : Flexible Schema
- SQL-Like
  - 99 % 표준 SQL과 동일한 Syntax
  - DML 지원 : Insert/Select/Update/Delete/Upsert
  - **INNER/OUTER JOIN 지원**

Power of SQL

Flexibility of JSON

```
{
    "Name" : "Jane Smith",
    "DOB"  : "1990-01-30",
    "Billing" : [
        {
            "type"    : "visa",
            "cardnum" : "5827-2842-2847-3222",
            "expiry"  : "2025-03"
        },
        {
            "type"    : "master",
            "cardnum" : "6274-2542-5847-3949",
            "expiry"  : "2024-12"
        }
    ]
}
```

4 데이터 모델 : 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

4 데이터 모델 : Time-Series

- **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 도큐먼트 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]
    ]
}
```

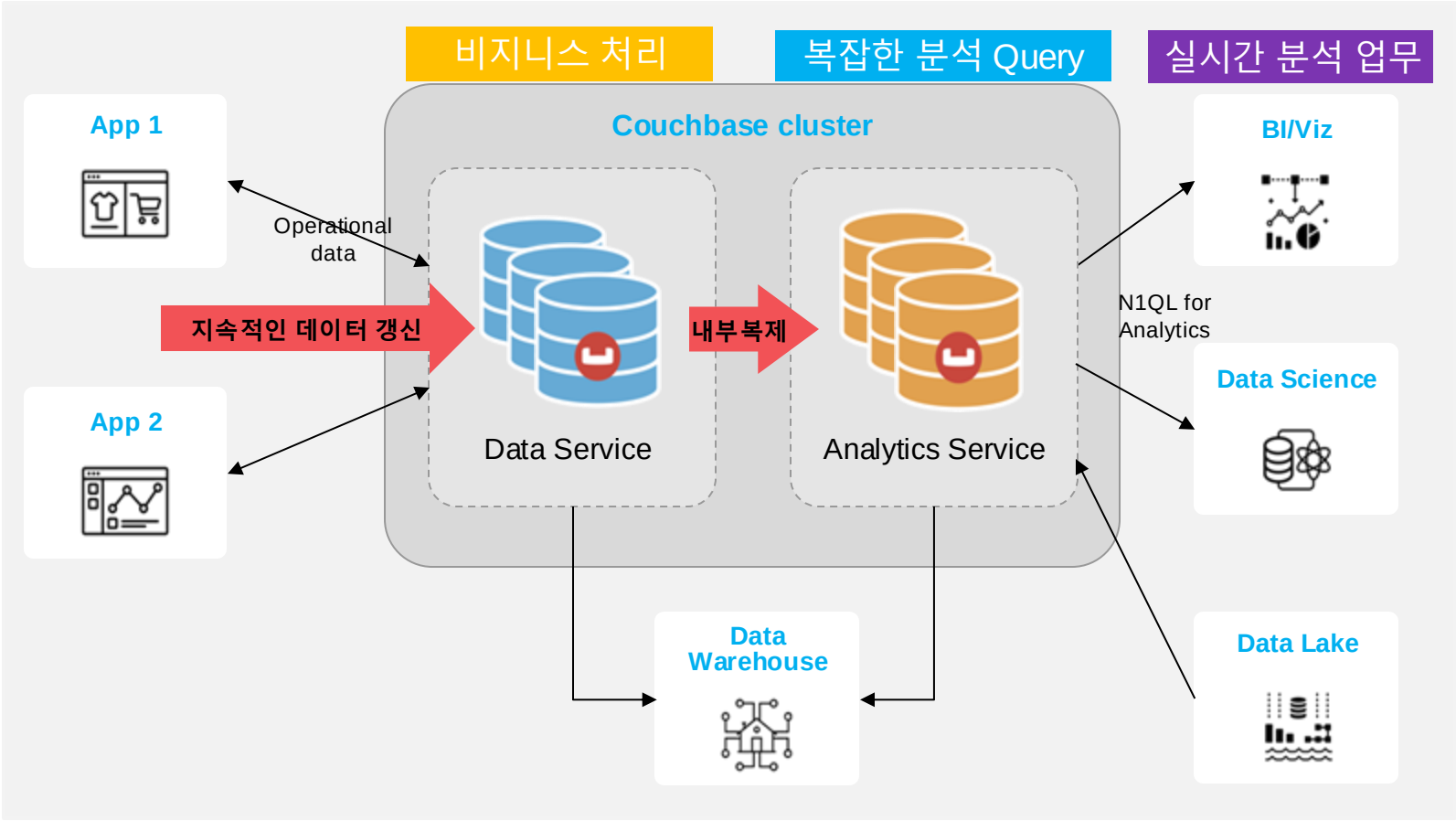
```
SELECT t.*
FROM Raw_Collection AS d
UNNEST _timeseries(d, {"ts_ranges":$ts_ranges}) AS t
WHERE d.equip_id = ‘1’
AND (d.ts_start <= $ts_ranges[1]
AND d.ts_end >= $ts_ranges[0]);
```

5 서비스 : Analytics > Hybrid Transaction Analytical 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

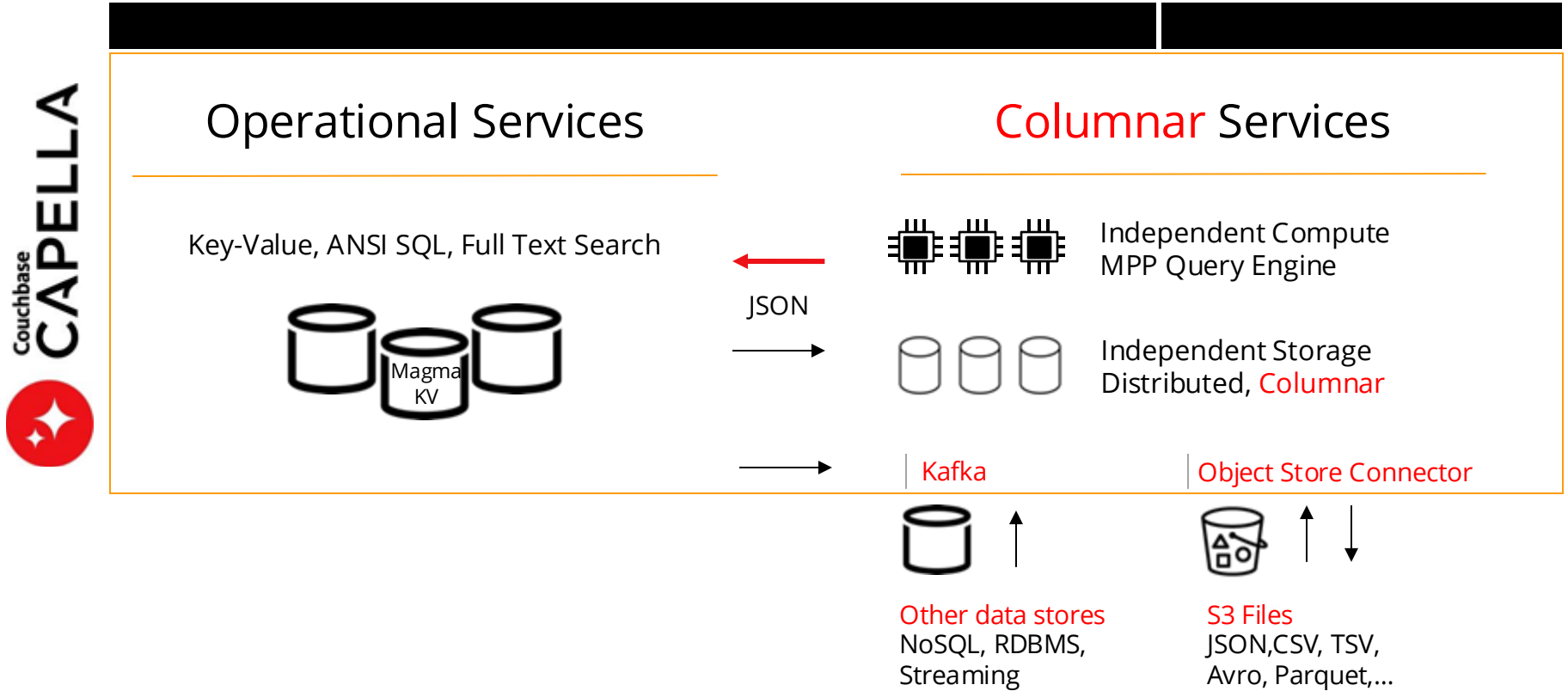
- **Parallel Processing**
  - SQL을 Analytics Service를 구성하는 노드에 분산하여 병렬 처리
  - 복잡한 Query 및 ad hoc Query 수행에 적합
  - 대용량 데이터 처리
  - Tableau Native Connector 제공
  - Power BI Native Connector 제공



5 서비스 : Capella Columnar Service : Cloud Data Lake

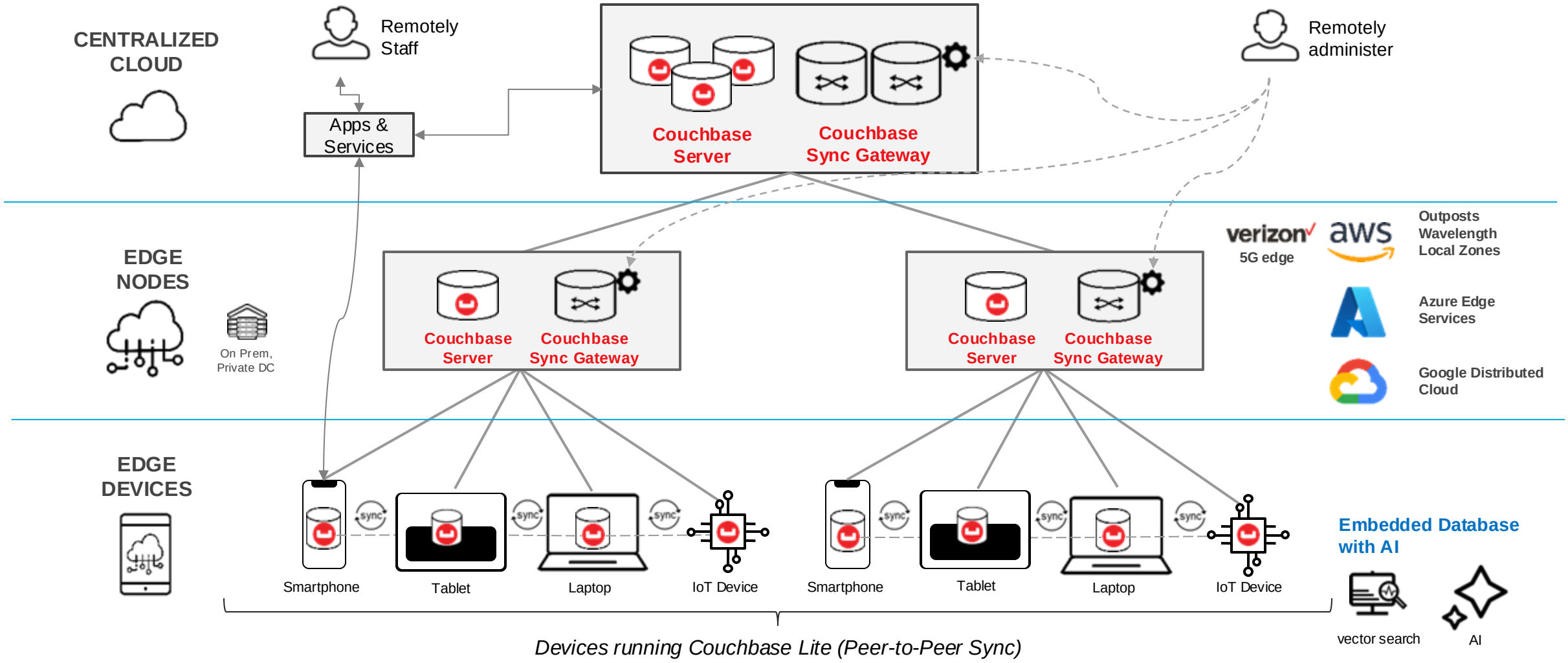
내부 JSON 데이터를 비롯하여 다양한 외부 데이터 소스를 통합하여 분석을 수행하는 컬럼 기반 Data Lake 기능 출시

Real-time Adaptive Applications (Operational, Transactions, Analytics, AI)



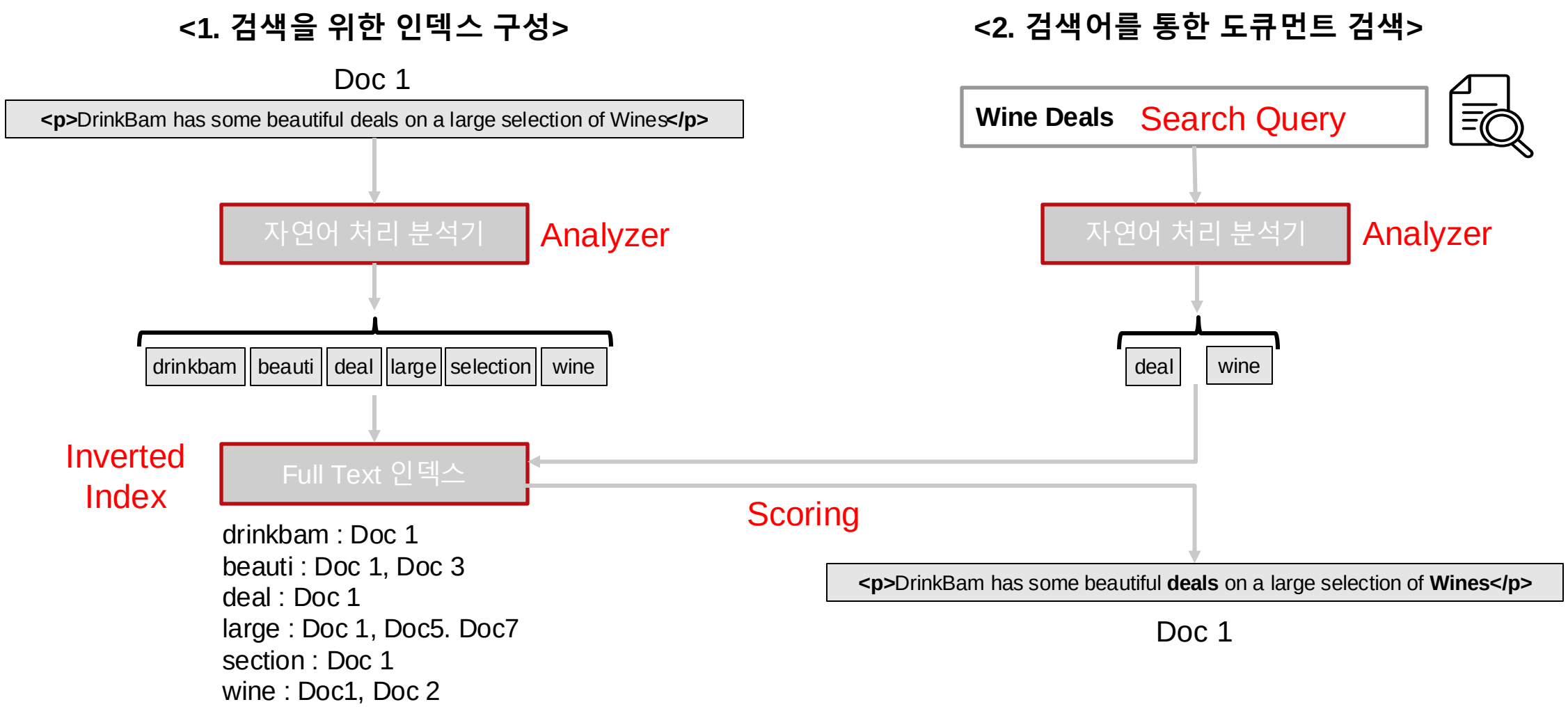
5 서비스 : Mobile (Couchbase Lite, Sync Gateway)

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



5 서비스 : Full Text Search(자연어 검색)

Data와 Query 서비스에 통합하여 검색 서비스 제공



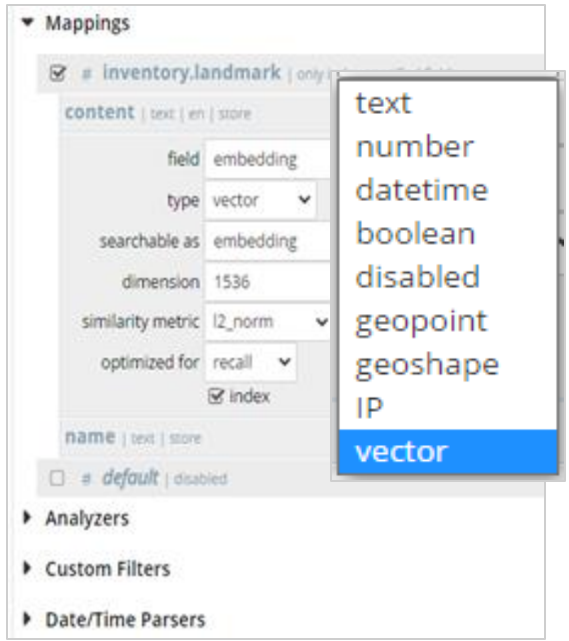
5 서비스 : Full Text Search with AI 벡터 검색

SQL 쿼리에 Full Text 검색 쿼리와 벡터 검색 쿼리를 통합 가능

JSON Storage

```
{ "type": “shoes",
 "productId": "CP123456",
 "category": “Gym Shoes",
 "name": “Beach Sneakers",
 "brand": “Ultimate Surf",
},
  "description": “The ultimate companion for beach adventurers, designed to seamlessly transition from sandy shores to urban landscapes. This innovative sneaker features a water-resistant, quick-drying mesh upper, allowing your feet to breathe while keeping them dry.",
 “descriptionVector": [0.131, 0.339, -0.611,0 .981,...]
```

Indexes



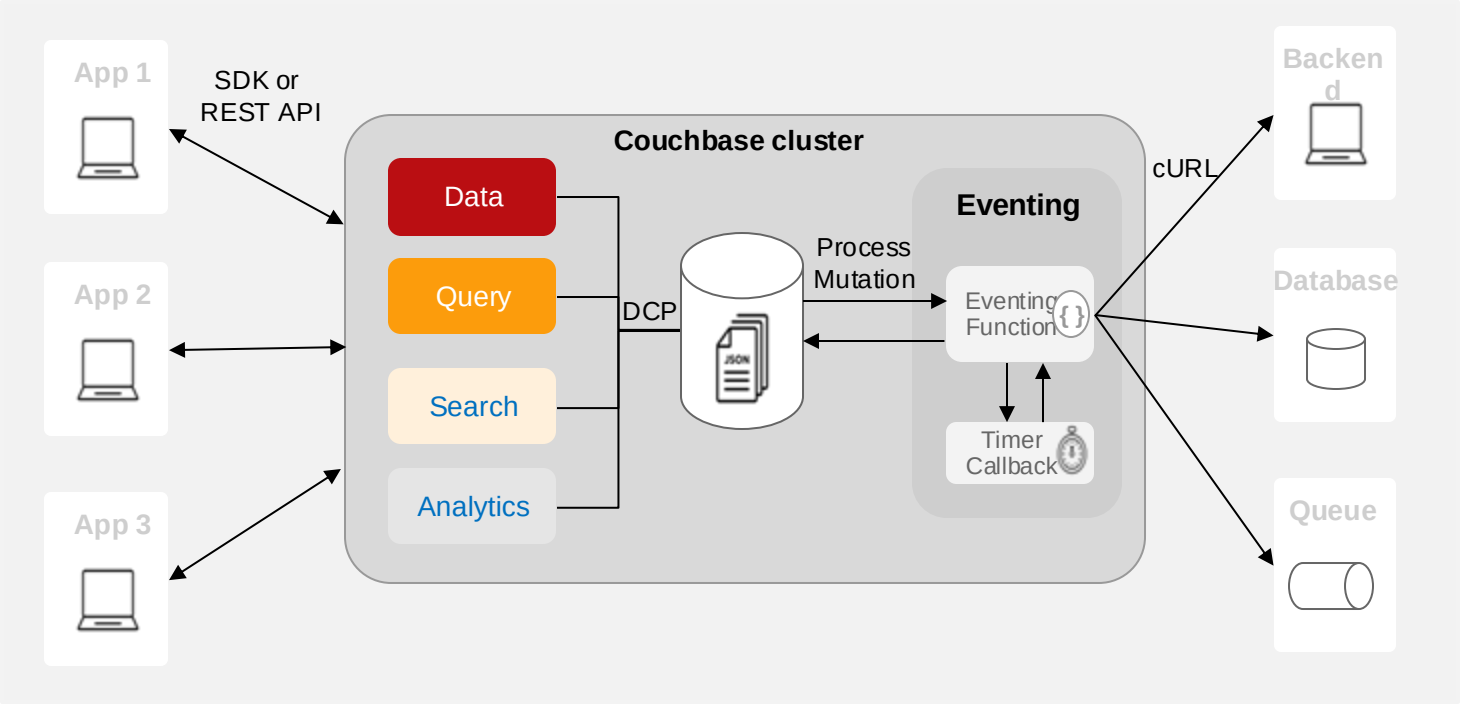
SQL + 자연어 검색 + AI Vector Search

```
SELECT meta().id, t.name
FROM `product` as t
WHERE
SEARCH(t,
 {“query”:{“match_phrase”: “sneaker”, “field”: “description”}}
)
ORDER BY search_score() DESC
LIMIT 10;
```

```
SELECT *
FROM product
WHERE LOWER(product.type) = ‘shoes’
AND product.size = 11
AND product.price between 50 and 80
/* desc SIMILAR TO  ‘blue running shoes’ */
ORDER BY GSI_VECTOR_ORDER(desc_embedding, {
  "knn": [{
    "field": "desc_embedding",
    "vector": [0.1, 0.334, -0.604, 0.985]  }]}
LIMIT 4
```

5 서비스 : Eventing

- 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

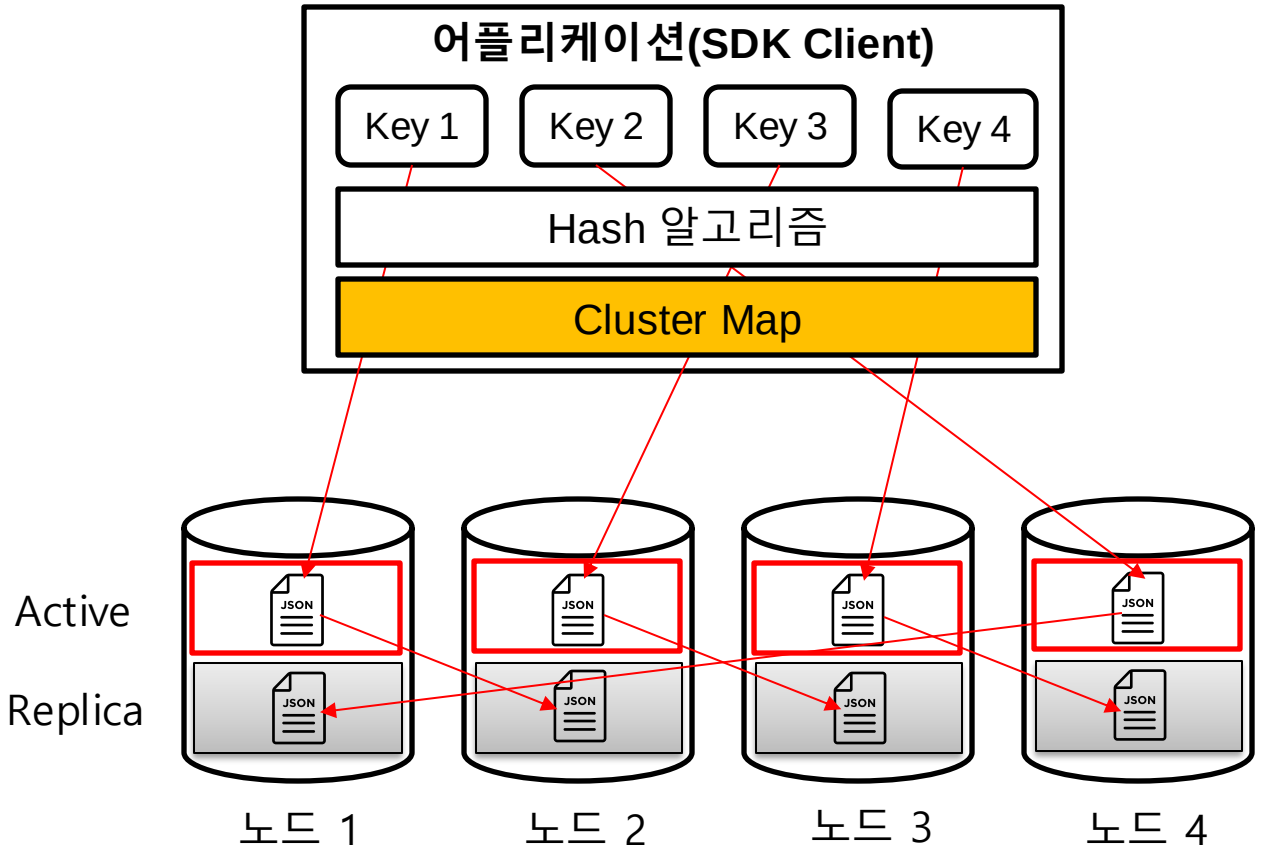


- **비즈니스 로직을 중앙 집중적을 관리**
  - Document 데이터 변경에 실시간으로 Trigger되는 Business Logic 구현
  - Java Script 기반으로 Cluster 내 Key-Value operation, SQL++ Query 지원
  - cURL function을 통해 외부 REST API 호출 가능

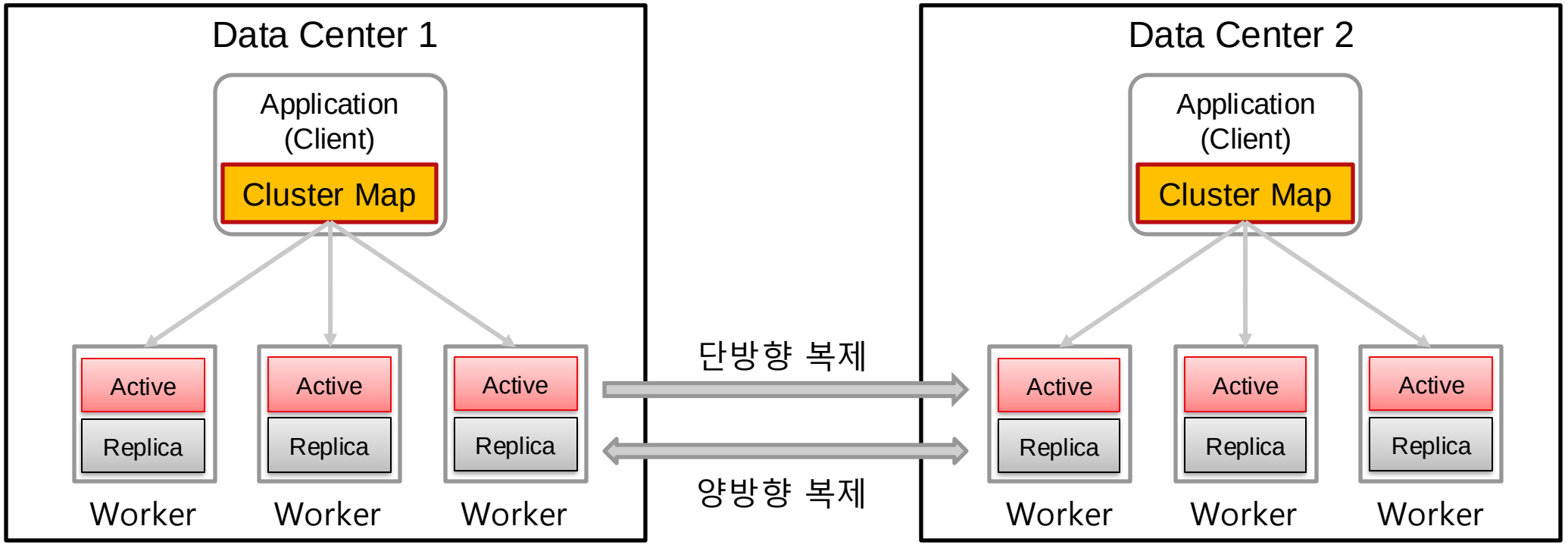
6 가용성 : 빠른 Fail-over를 통한 고 가용성

- **Replica 버켓**
  - Auto-sharding을 통한 균등한 데이터 분산
  - Active 버켓에 대한 Replica 버켓을 내부 노드 단위에서 관리
  - Replica 버켓을 최대 3 까지 가능

- **Couchbase**
  - 특정 노드의 장애가 발생하면 장애가 발생한 Active 버켓의 Replica 버켓을 Active 상태로 전환
  - 빠른 Fail-over
  - 별도의 Replica 노드나 Standby 노드가 필요 없음



6 가용성 : 클러스터 간 Native 복제를 통한 재해 복구



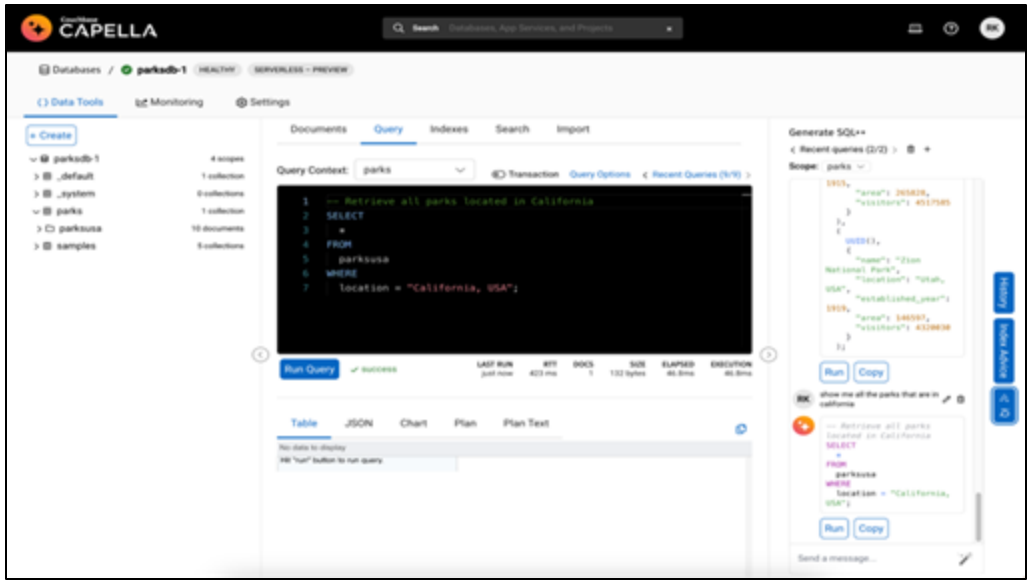
- **XDCR(Cross Data Center Replication)**
  - XDCR을 통해 멀티 데이터 센터에 위치한 클러스터 간 데이터 복제
  - 단방향 복제 및 양방향 복제 지원
  - 복제는 필요한 데이터만 필터링 가능
  - 단순 재해 복구 솔루션 이상의 글로벌 워크로드 분산 솔루션

7 개발 편의성 : 생성형 AI 기반 코딩 지원

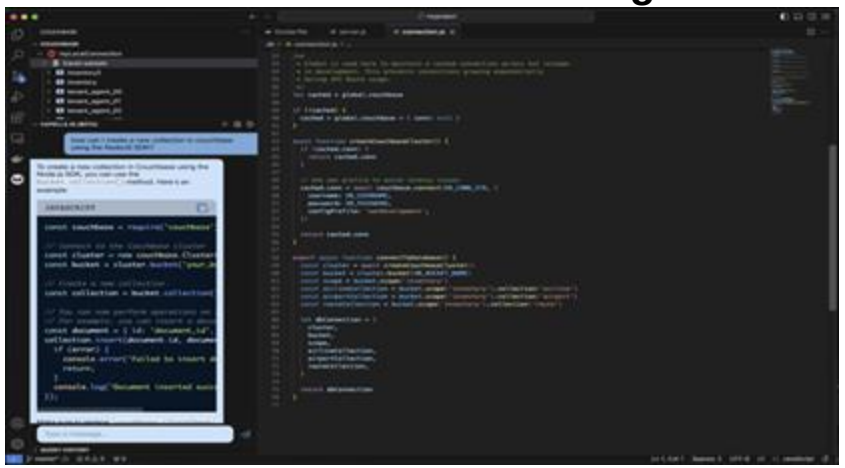
- Generative AI의 LLM을 활용한 Couchbase Capella 전용 Code Assistant
- 자연어로 SQL 및 소스 코딩 지원
- Couchbase 내부 스키마 정보를 활용하여 실제적인 코딩 지원

Couchbase CAPELLA | iQ

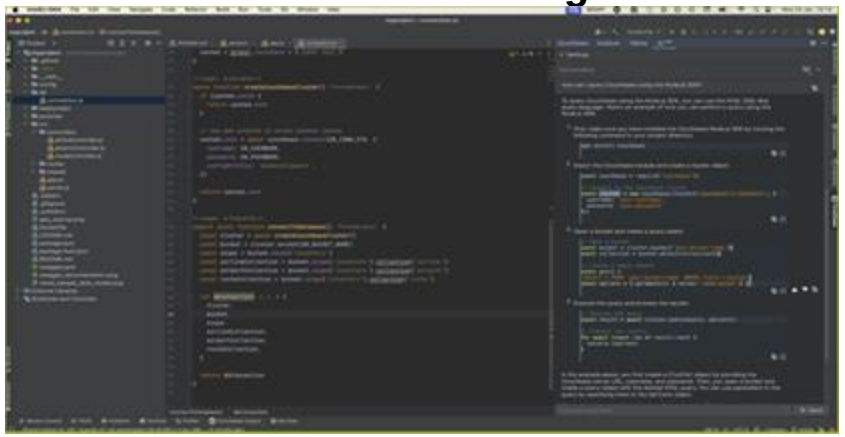
Capella Workbench



Visual Studio Code Plug-in



IntelliJ IDEA Plug-in

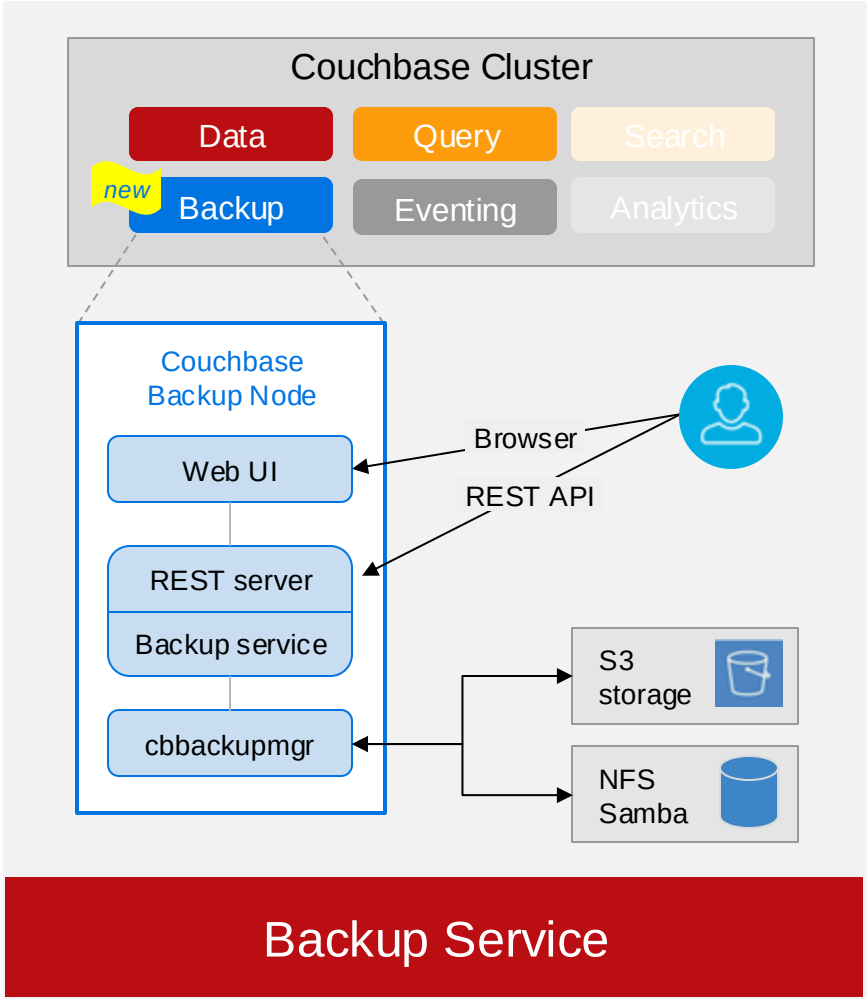


8 운영/관리 : 내장 백업복구 솔루션

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

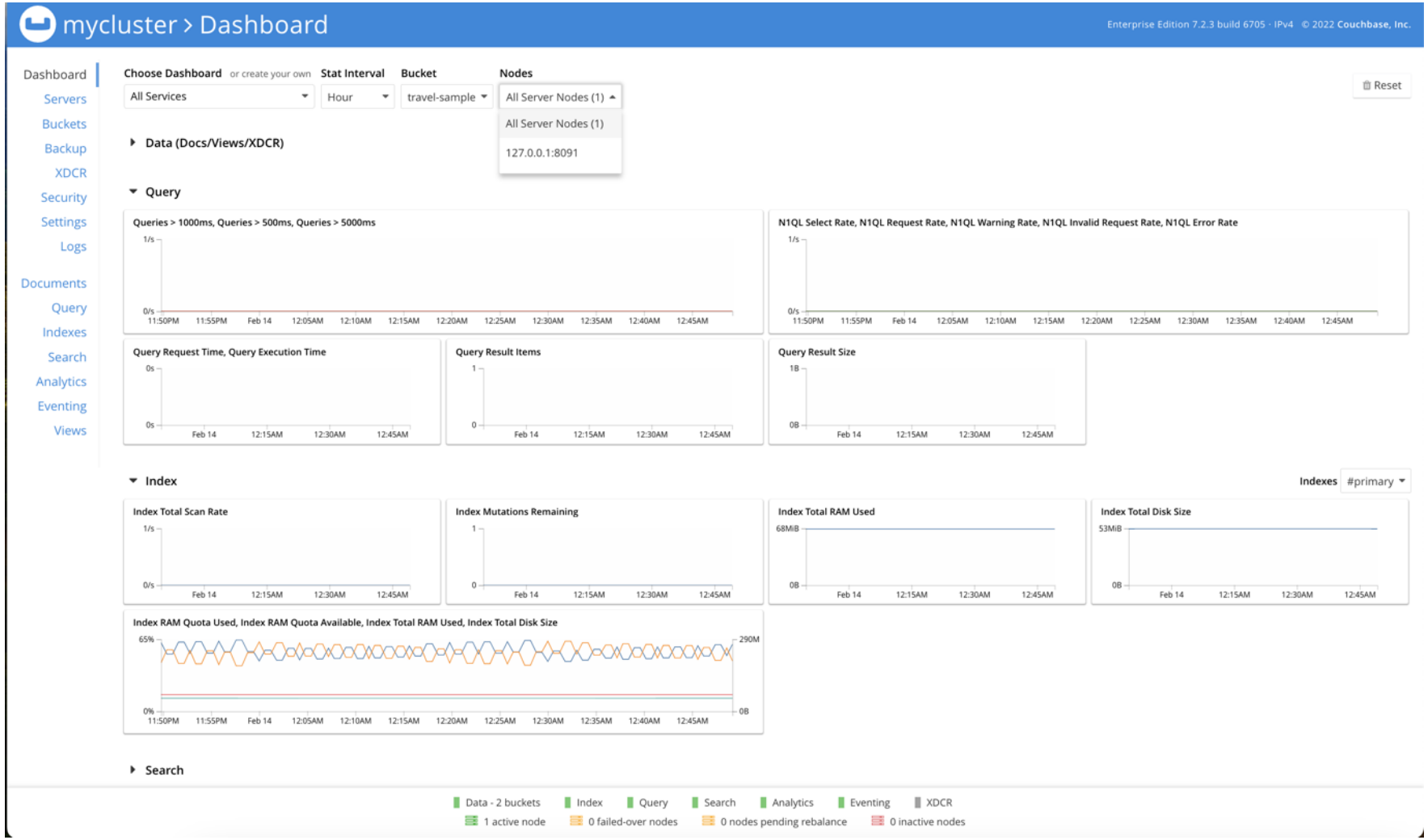
- **Backup 방식**
  - Full Backup
  - Incremental Backup
  - Merge Backup File

- **Restore 방식**
  - Bucket, Scope, Collection 단위 가능



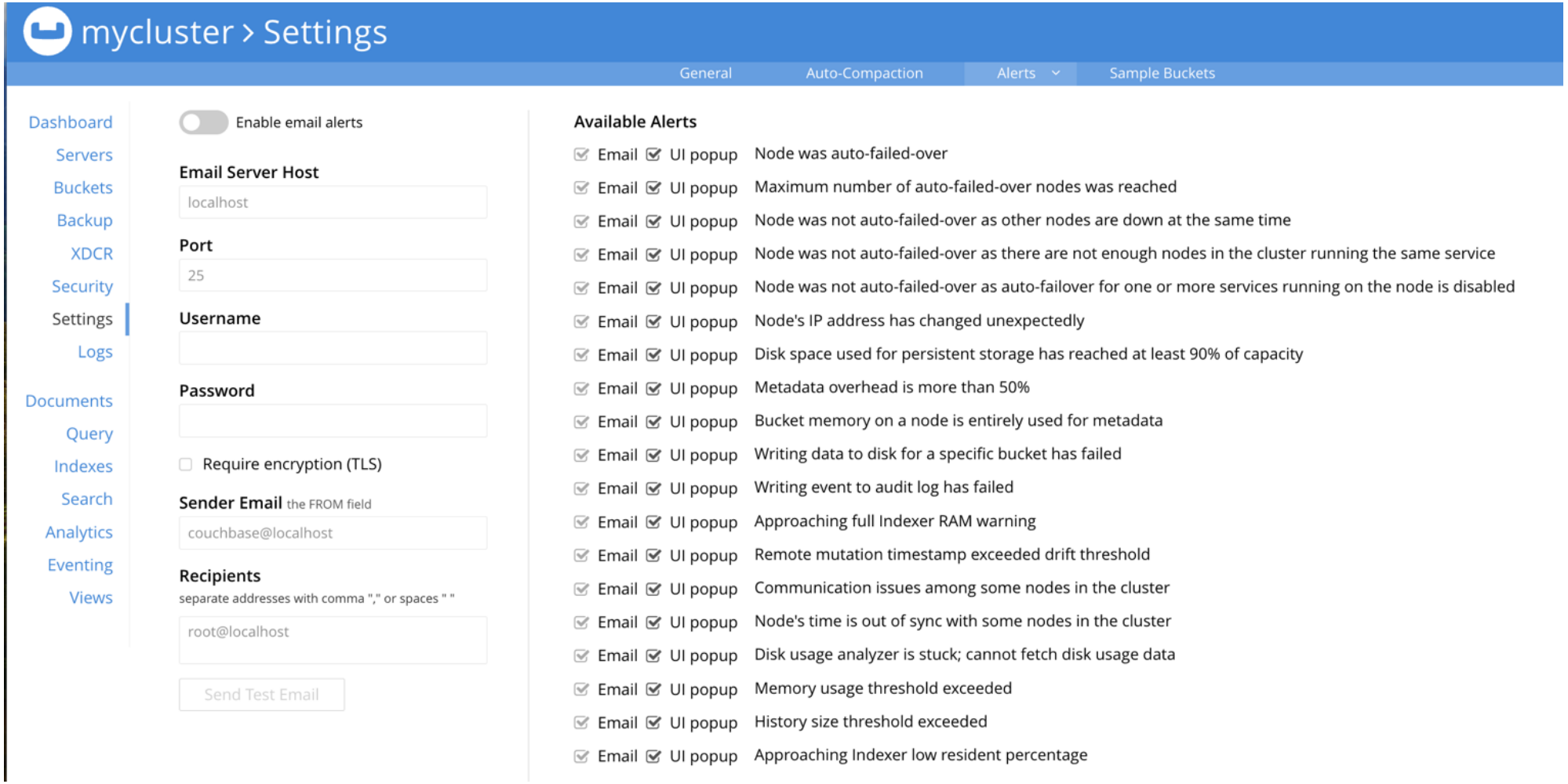
8 운영/관리 : 성능 모니터링

Couchbase는 다양한 성능 매트릭을 제공합니다. 그리고, Prometheus/Grafana 로도 관리할 수 도 있습니다.



8 운영/관리 : 시스템 상태 경고 및 권고

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



9 보안 > End-to-End 보안

Couchbase는 엔드 투 엔드 보안 기능을 제공합니다. 클라이언트 접속 인증에서 네트워크 보안, 서버 스토리지 저장시 암호화할 수 있으며, 다양한 감사 기능 제공합니다.

Authentication 인증

- App/Data: SASL Authentication
- Users Database:
  - Local
  - LDAP / AD
  - PAM

Authorization 권한

- RBAC
  - For Admin
  - For Data
- Limits at various levels
  - Cluster
  - Bucket
  - Scope (v7.0)
  - Collection (v7.0)

Crypto 암호화

- TLS Admin Access
- TLS client-server access
- TLS XDCR
- X.509 certificates
- Secret Management
- TLS Protocol and Cipher Mgnt
- Data-at-rest Encryption*

Auditing 감사

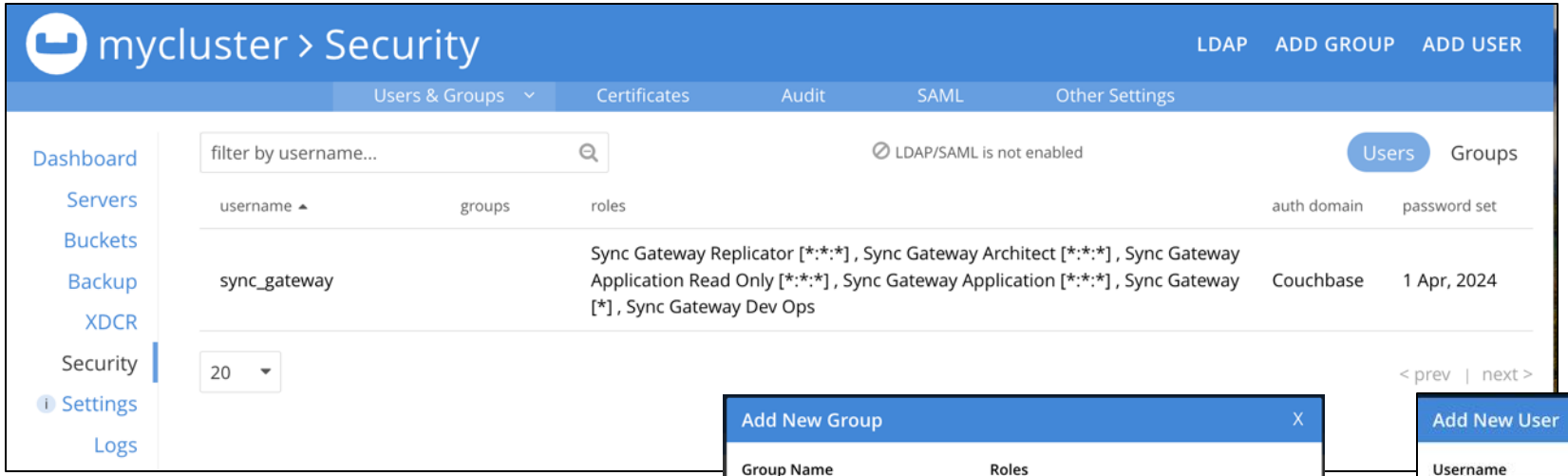
- Audit Events:
  - Login
  - Add node
  - Settings
  - N1QL
  - Etc.

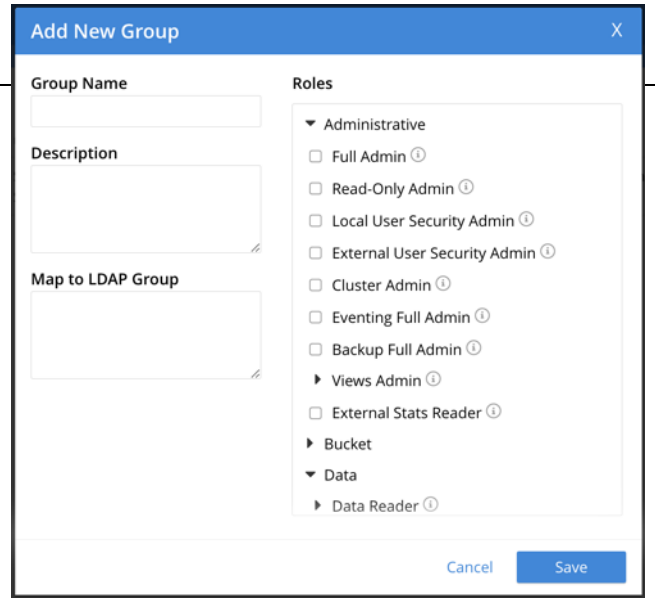
Operations 운영보안

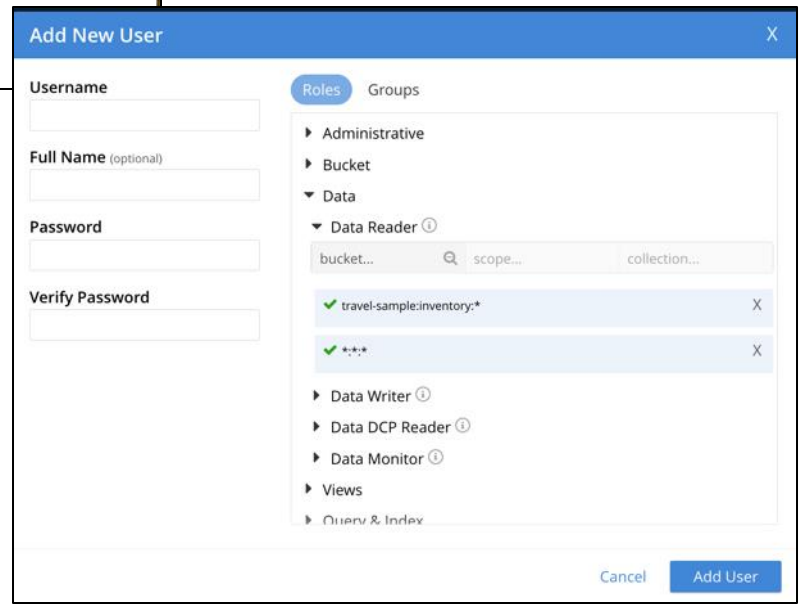
- Security management via UI/CLI/REST
- Log Redaction
- Non-Root Install

9 보안 > 사용자/그룹 권한 관리

Couchbase는 그룹과 사용자에 대한 권한 관리를 통해 Admin 관리 기능과 데이터별 쓰기/읽기를 허용할 수 있습니다.







10 설치/구성 : 지원 플랫폼

Bare-Metal, VM, Container 와 완전관리형 클라우드 데이터베이스 서비스(DBaaS) 사용 가능

Couchbase CAPELLA

Couchbase

Fully Managed

- **완전관리형 데이터베이스 서비스**
- **AWS, GCP, Azure**

- **설치, 구성, 모니터링, 업그레드 등 모든 운영은 Couchbase가 담당**

Enterprise

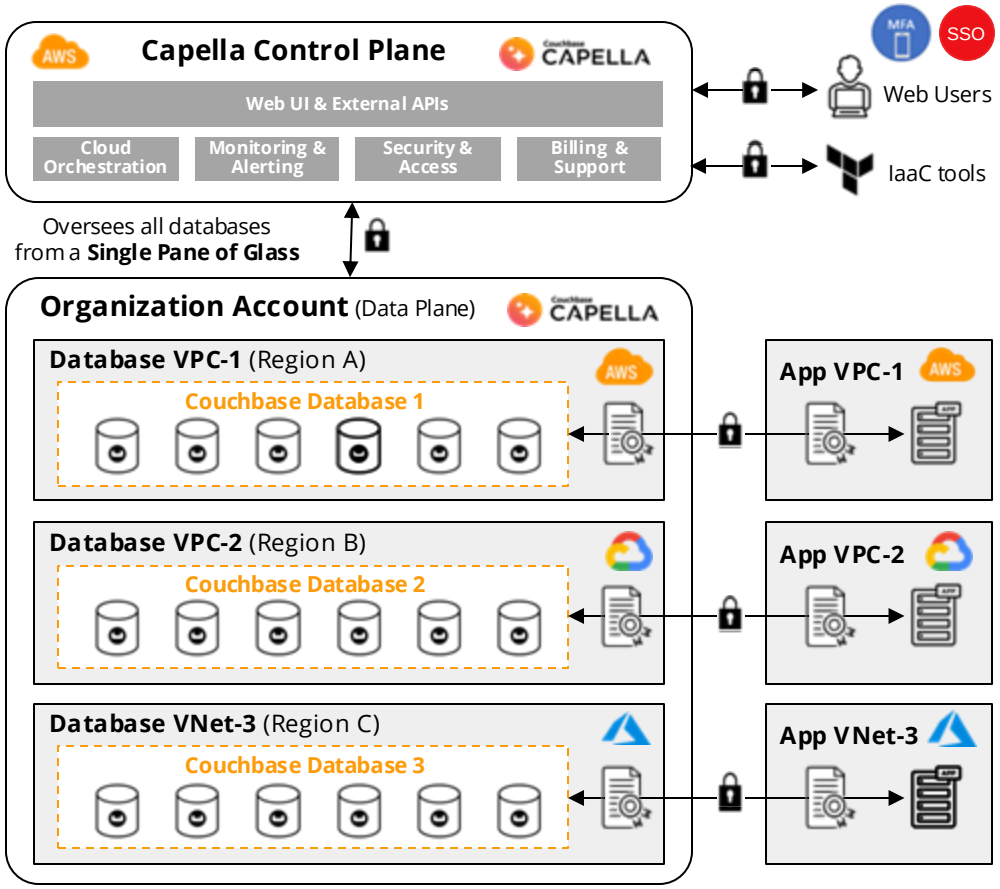
- Bare-metal 서버, 클라우드 IaaS 서버
- Private Cloud 서버, K8S 컨테이너

- 설치, 구성, 모니터링, 업그레드 등 모든 운영은 고객이 수행

Linux, Windows, MacOS, Intel/AMD, ARM

10 설치/구성 : 카펠라 아키텍처

완전관리형 클라우드 데이터베이스 서비스(DBaaS)인 Capella 아키텍처



Capella Control Plane

- **Manages** the Cloud Orchestration, Monitoring & Alerting, Security & Access, Billing & Support
- Is the Access Point for Organization **Web UI** Users
- Allows Infrastructure as Code Tools (e.g. **Terraform**)

Organization Account (Data Plane)

- **Account Isolation**: 1 Account per Organization
- **Database Isolation**: 1 VPC per Couchbase Database
- **Multiple Clouds**: AWS, Google and Azure

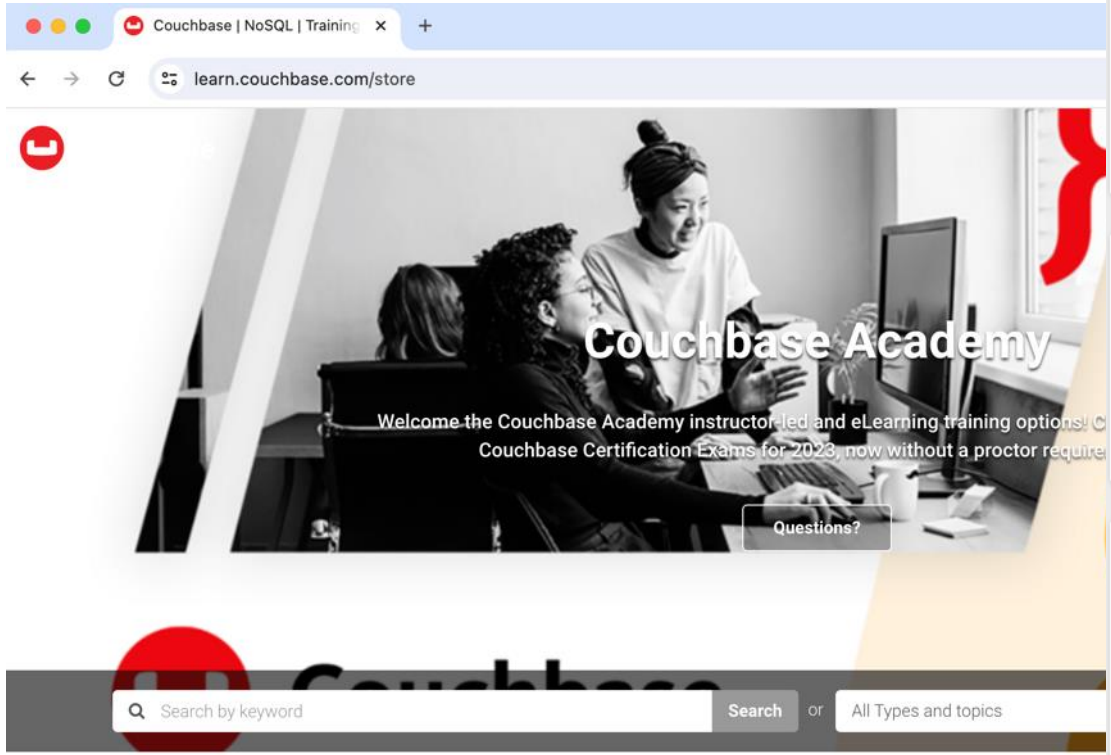
Applications

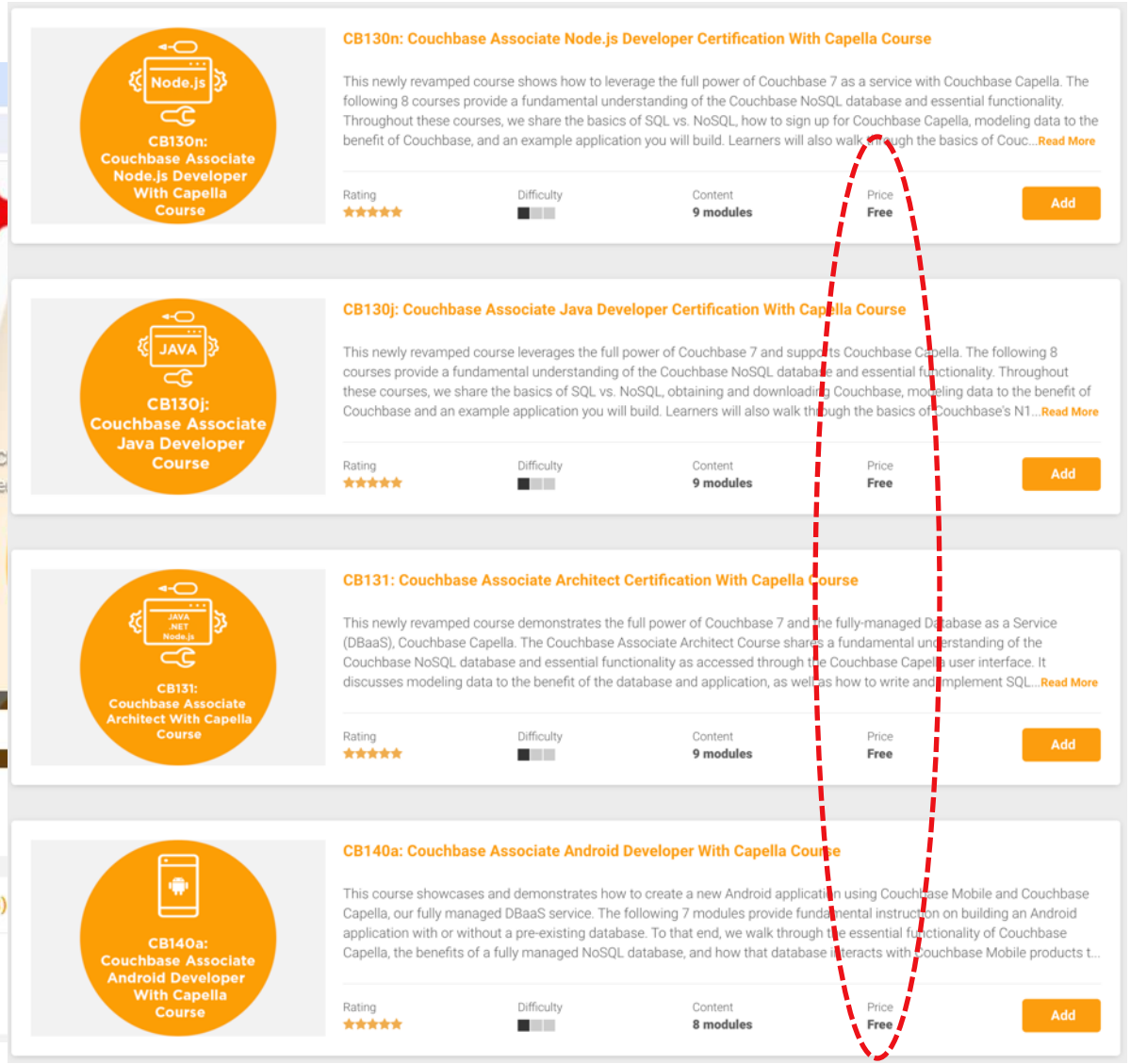
- **Connect** directly to Databases
- **Multiple** connectivity options: over Public Connection, through VPC Peering or Private Link
- All communications are **encrypted**

11 교육 : Couchbase Academy

https://docs.couchbase.com/
https://developer.couchbase.com/
https://couchbase.live/
https://query-tutorial.couchbase.com/tutorial/#1

https://learn.couchbase.com/store





12 요약하면,

Couchbase는

JSON Document 직관적이고

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

SQL, Generative AI 익숙하고, 쉽게

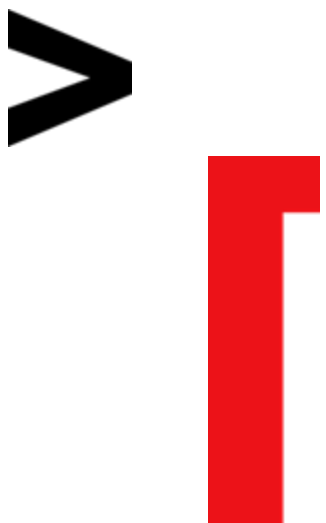
- NoSQL 이지만 표준 SQL을 지원
- 생성적 AI인 Capella IQ 지원으로 더 손쉬운 개발이 가능

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

A Brief History of AI



Key Milestones in the History of AI | 20th Century

AI is born | **Early Developments** | **Expert Systems** | **Machine Learning**

Can a Machine think?

1950

Alan Turing test - if a machine tricks a human into thinking it's a human, then it's intelligence.

AI term coined

1956

John McCarthy introduced the term 'Artificial Intelligence' at The Dartmouth Conference.

Gen. Problem Solver

1957

General Problem Solver is the first AI algorithm. It can solve formalized problems (e.g. Towers of Hanoi).

The chatbot Eliza

1965

Eliza is the first chatterbot - or "chatbot" modernly. It can simulate a human conversation.

R1, an Expert System

1980

R1 uses 2500 rules to ensure that the customer's order is complete, saving the company $25M a year.

Deep Blue

1997

IBM's Deep Blue supercomputer defeats the world chess champion Garry Kasparov.

Kismet

1998

Cynthia Breazeal at MIT creates Kismet, a robot that can detect and respond to people's feeling.

1st AI Winter
1974-1980

2nd AI Winter
1987-1993

Key Milestones in the History of AI | 21st Century

Machine Learning | Deep Learning | Generative Models

Watson

2006

IBM's Watson question-answering system is created. In 2011, it defeats Jeopardy!'s champion.

Siri & Alexa

2011

Apple released Siri, an AI assistant that can respond to voice. 3 years later, Amazon launched Alexa.

Google AI

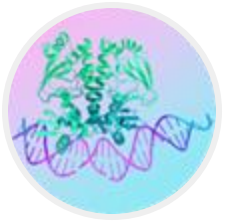
2014

Google AI recognizes cats from 10 million Youtube videos with almost 75% accuracy in 3 days.

AlphaGO

2016

AlphaGo defeats top Go player Lee Sedol in Seoul. Go is incredibly difficult given the vast number of positions.



AlphaFold

2020

DeepMind's AlphaFold system can predict protein structure, with implications for drug discovery and biology.

AI Boom

AI Explosion

DALL.E

2021

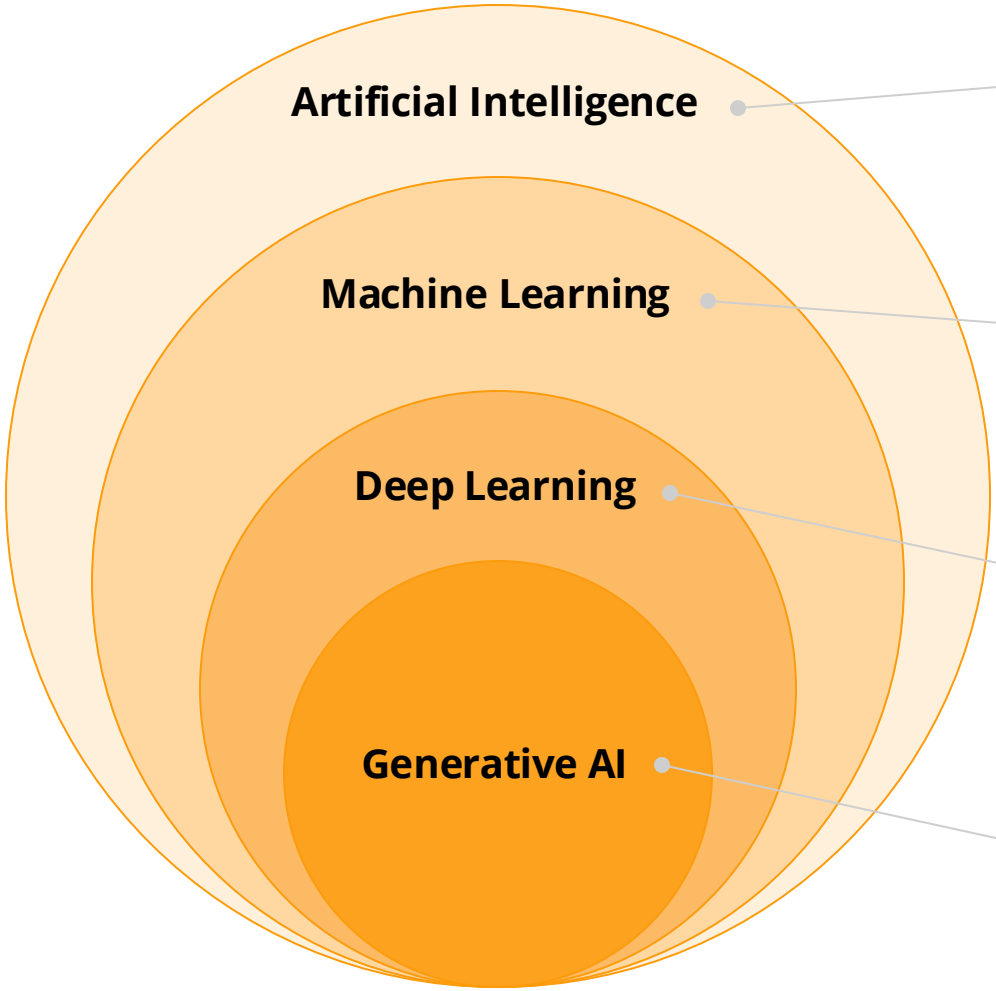
DALL·E, and a year later Midjourney and StableDiffusion, can generate images from textual descriptions.

ChatGPT

2022

OpenAI releases the AI chatbot ChatGPT. It is based on the Large Language Model GPT-3 created in 2020.

The Technology behind AI



Artificial Intelligence (AI)
Techniques that allows computers to emulate human behavior (e.g. learn, recognize patterns, solve complex problems).

Machine Learning (ML)
A subset of AI, using advanced algorithms to detect patterns in large data sets, allowing machines to learn and adapt for prediction or content generation use cases.

Deep Learning (DL)
A subset of ML, using multiple layers of artificial neural networks that simulate human brains for in-depth data processing.

Generative AI (GenAI)
A subset of DL, using models that generate content like text, images, or code based on provided input.

Powering Apps: A Combination of Predictive & Generative AI

Predictive AI

Outcomes and Insights driven by ML

- Predict Outcomes based on historical data
- Utilize ML algorithms for pattern recognition
- Learns patterns and correlations from data
- Drives decision making and Future planning
- High ROI, trained on proprietary data

- Predictive Insights
- Dynamic Pricing
- Fraud Detection
- Inventory Optimization

+

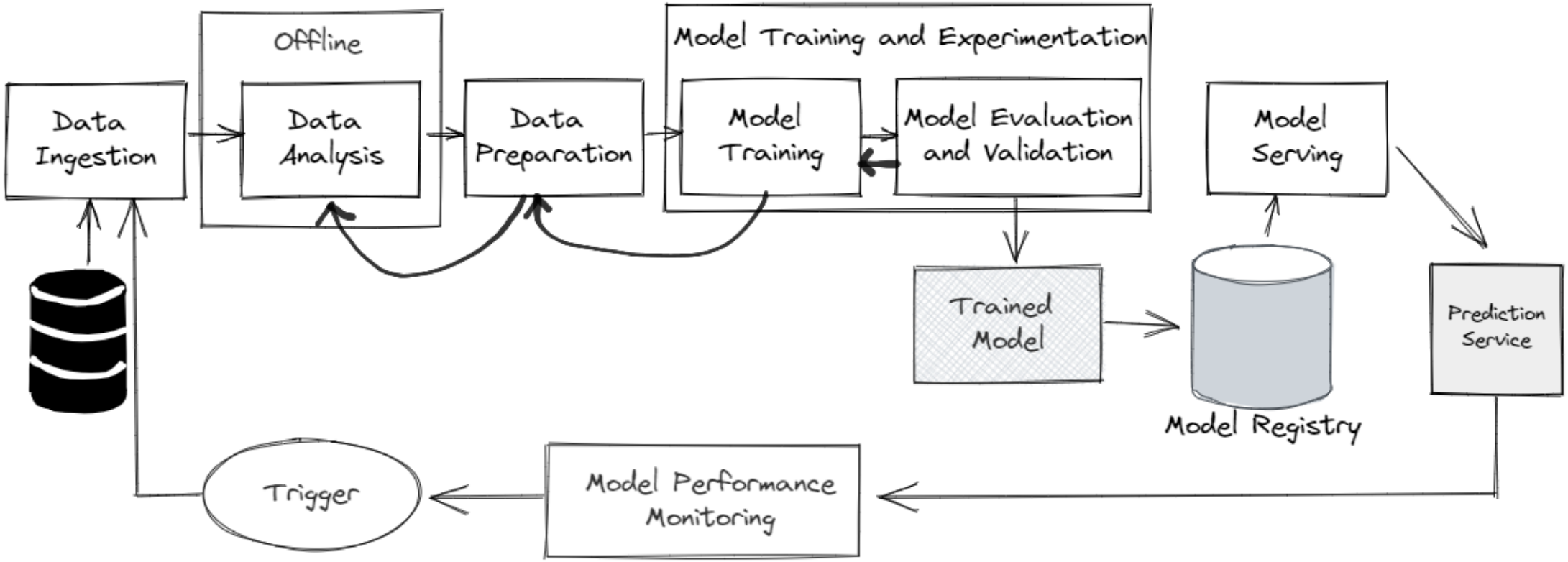
Generative AI

Generate Content and Experiences

- Generate or Synthesize content
- Needs large amounts of unlabeled data for training
- Generates new data probabilistically
- Fosters creativity, innovation
- Accelerates human productivity

- Hyper-personalized experiences
- Contextualized content
- Chatbots and CoPilots
- Synthetic data and Summarization

Model? Machine Learning Workflow



출처 : https://www.iguazio.com/blog/ml-workflows-what-can-you-automate/
https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

Vector DB, Search

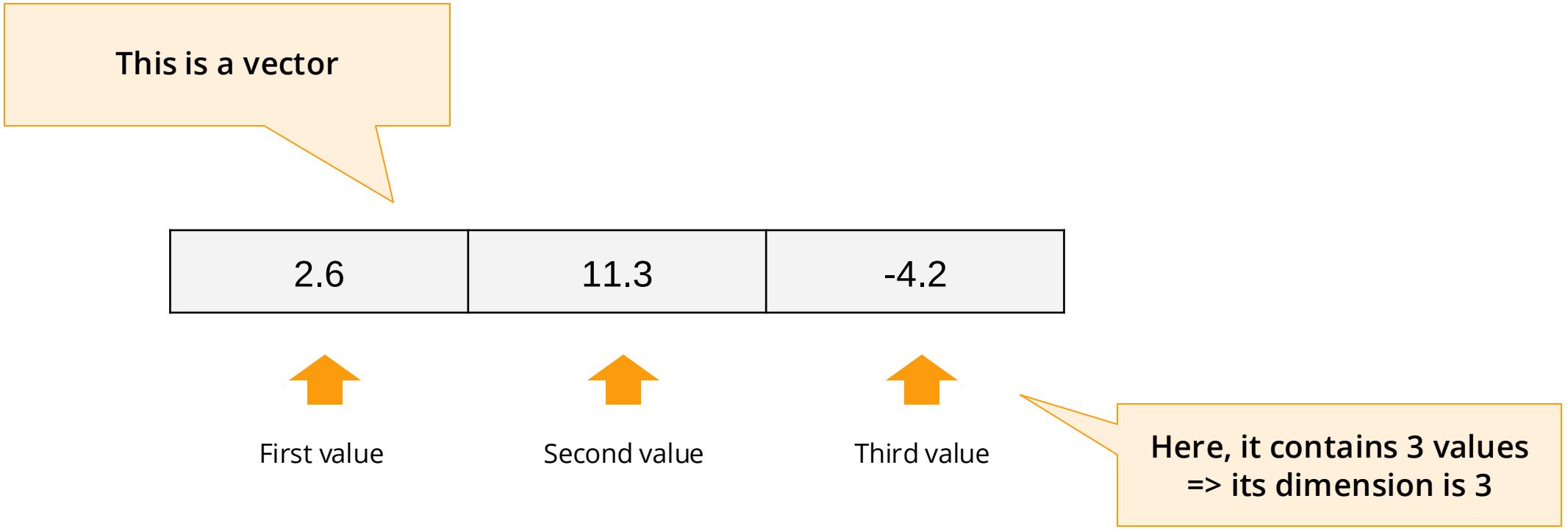
1. Demystifying Vector Search
2. The power of Hybrid Search

GenAI (LLM, RAG)

3. Vector Search Use Case: Semantic Search
4. A quick tour of LLMs and Generative AI
5. Vector Search Use Case: Retrieval-Augmented Generation (RAG)

Demystifying Vector Search

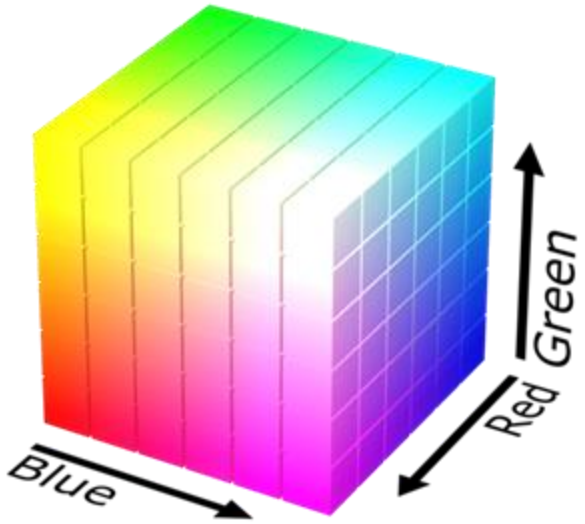
What is a Vector



A Vector is a just an **array of numerical values**

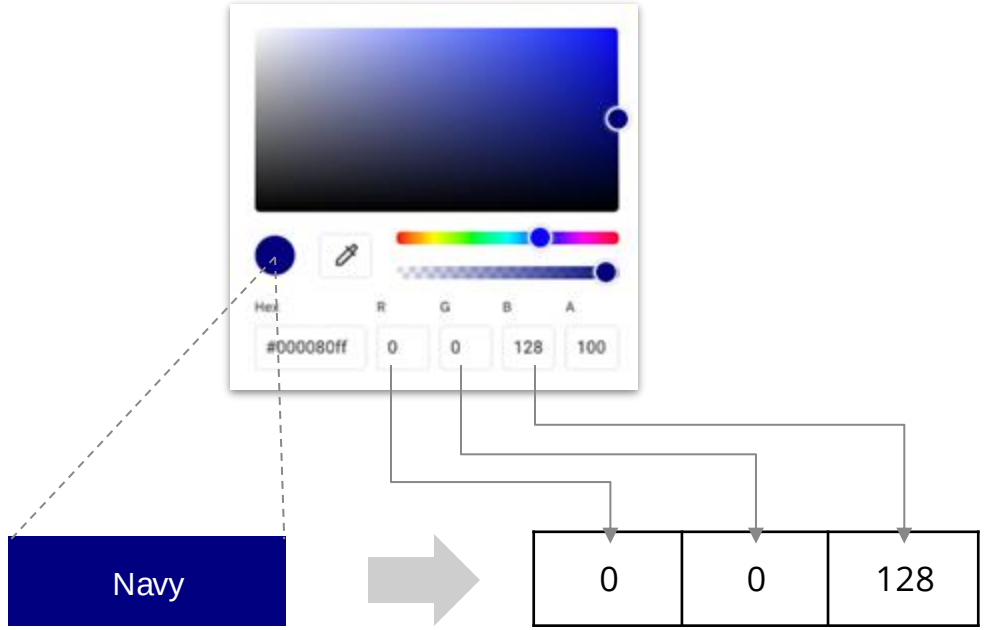
What is a Vector used for | Basic RGB Example

The RGB model example



The model used to create the colors you see on TV and computer screens.
Each color is the addition of a **R**ed, **G**reen and **B**lue primary colors.

A Vector representation of a Color



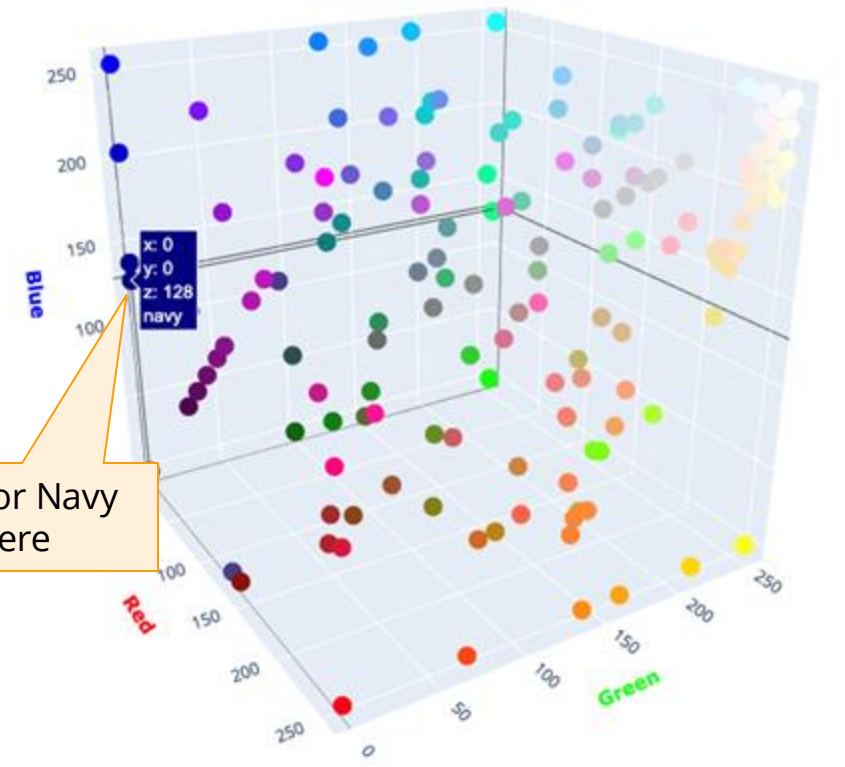
The color *Navy* ...

... is represented in the RGB model with this vector

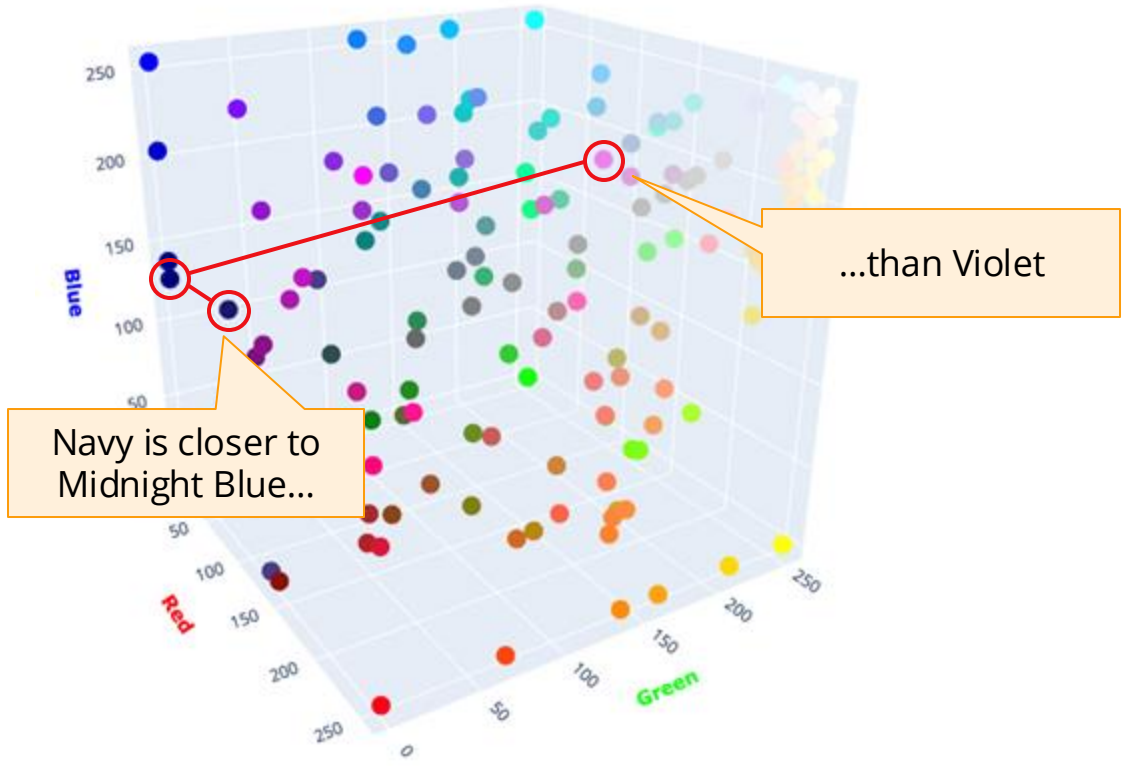
A **Vector** is used to **represent** a more complex object. The **Model** defines the **meaning** of each dimension.

Vectors Similarity

Example of 123 vectors of RGB colors



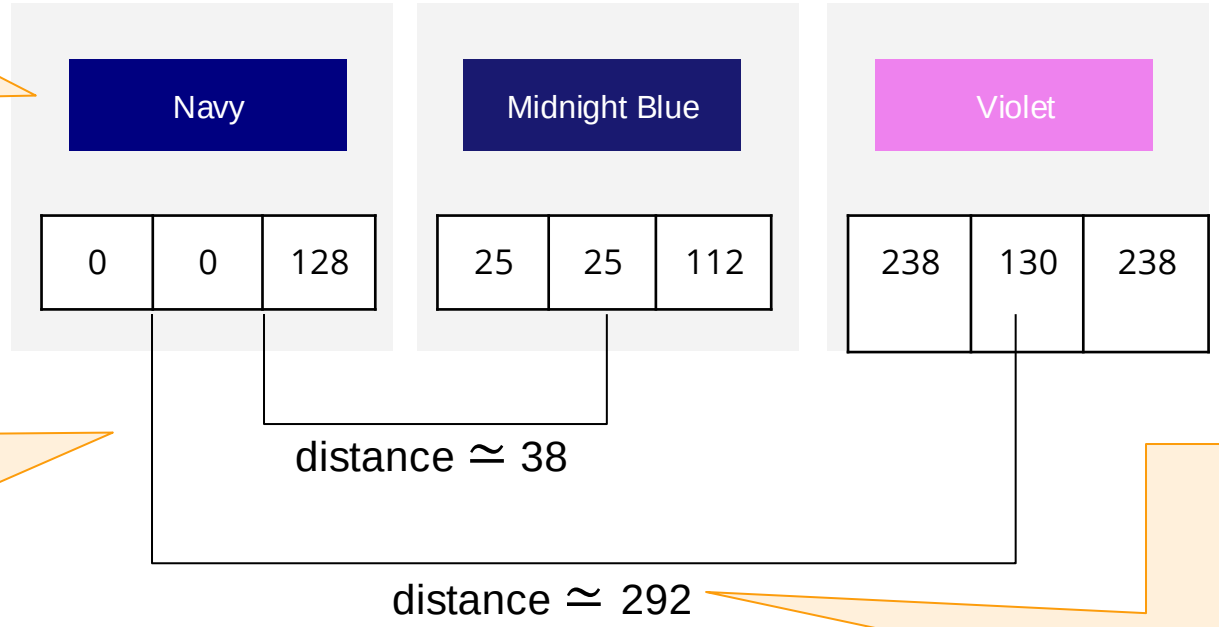
Similar colors are closer to each other



Vectors make it possible to translate **similarity** as perceived by humans to **proximity in a vector space**.

How does Similarity works

To the human eyes,
Navy is closer to Midnight Blue than Violet



Mathematically,
we got the same result by comparing the vectors

Vectors are compared using **a similarity distance.**

Here the *euclidean distance*

$292 \simeq \sqrt{(238-0)^2+(130-0)^2+(238-128)^2}$

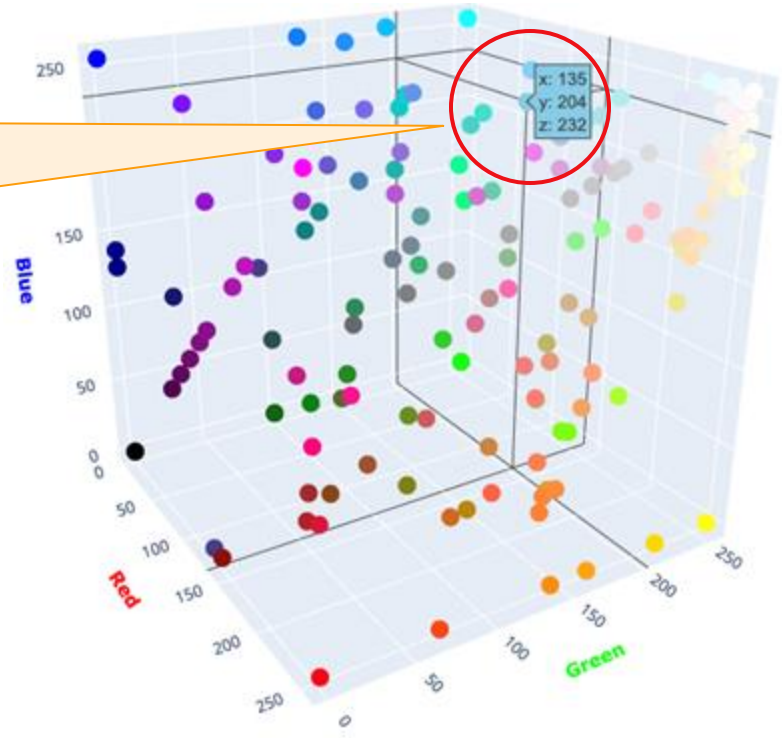
Vectors can easily be compared mathematically using a **similarity distance**

Similarity Search with K-NN (K-Nearest Neighbors)

Which are the top k nearest neighbors to this color?

[135,204,232]

Example of 123 vectors of RGB colors

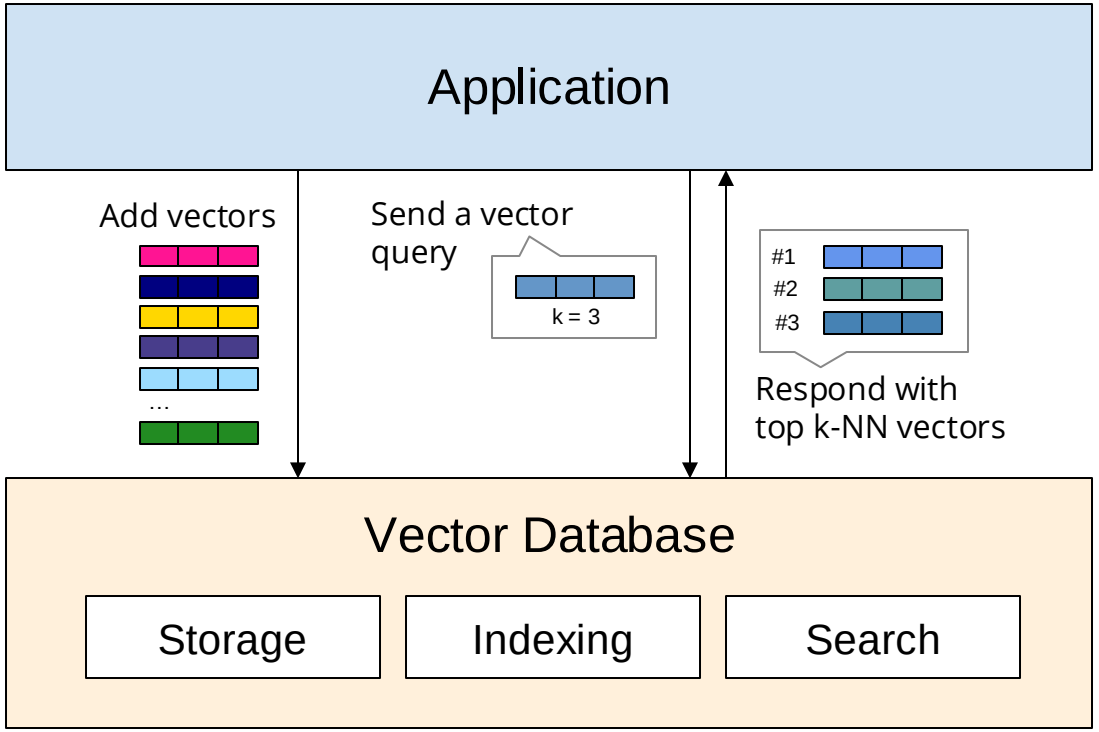


Top k-NN results of the query

#1	sky blue	[135,206,235]
#2	light sky blue	[135,206,250]
#3	light blue	[173,216,230]

A similarity search is a query that **finds the k nearest neighbors to a vector**, as measured by a similarity metric

What is Vector Database



Vector databases provide the ability to **store, index and search vectors** using similarity search

Couchbase Vector Search

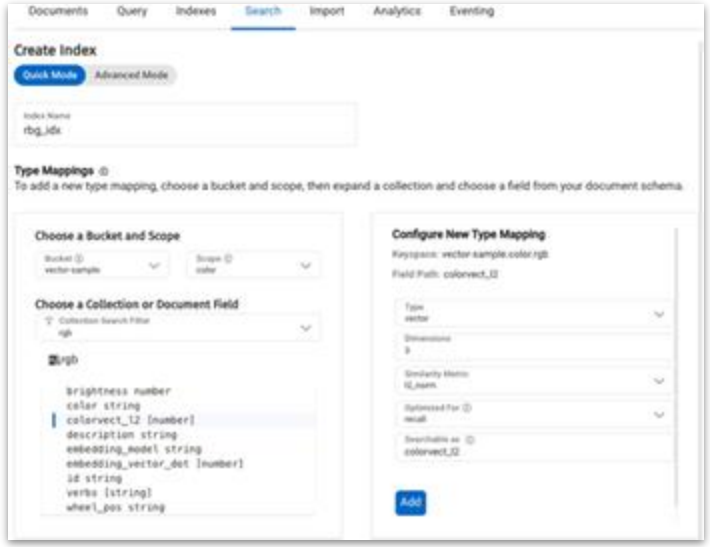
The vectors are stored as **a field in JSON** documents

```
{
 "id": "#000080",
 "color": "navy",
 "brightness": 14.592,
 "colorvect_l2": [0, 0, 128],
 "description": "Navy is a deep, rich color that
     exudes sophistication. It is a dark shade of
     blue that is often associated with authority,
     stability, and elegance. Navy is a versatile
     color that can be both bold and understated,
     making it a popular choice in fashion and
     interior design. It is a timeless color that
     never goes out of style and adds a touch of
     sophistication to any look or space.",
 }
```

JSON Storage

Data Service

A **Vector Index** must created to allow the vectors to be searched



Vector Index

A **Vector Query** can now search for the top k-NN of a color

```
{
  "query": { "match_none": {} },
  "knn": [
    {
      "field": "colorvect_l2",
      "vector": [135, 204, 232],
      "k": 3
    }
  ],
  "fields": ["color"]
}
```

Vector Query

Search Service

Couchbase uses the **Data Service to store vectors**, and the **Search Service to index and query vectors**

The power of Hybrid Search

Hybrid SQL++ and Vector Search with Couchbase

This is a SQL++ query

Combining Vector Search query

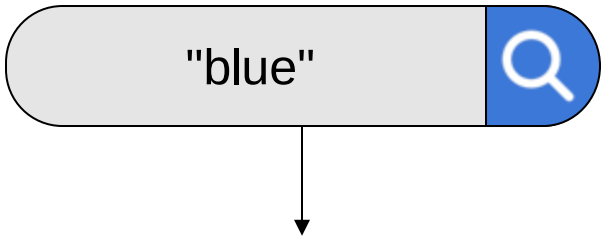
And standard SQL++ criteria

```
SELECT color, brightness
FROM `vector-sample`.color.rgb AS t1
WHERE
    SEARCH(t1,
     {
       "query": {  "match_none": {} },
       "knn": [{
         "field": "colorvect_l2",
         "vector": [135,204,232],
         "k": 3 }]
     }
AND
    brightness >= 180 AND brightness <= 190
    )
```

- ✓ SQL++ is easy and familiar to developers
- ✓ You can filter vector search results with other criteria
- ✓ **You don't have to run 2 separate databases, one for Documents and one for Vector Search!**

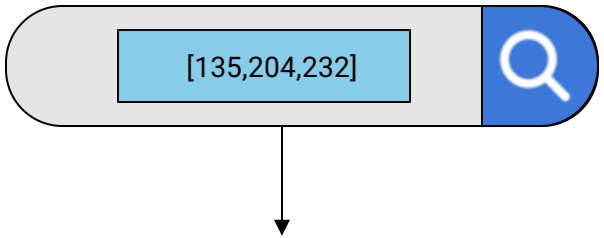
Couchbase can run hybrid SQL++ and Vector Search queries to **facilitate application development**

Comparison between Keyword Search and Vector Search



Keyword Search on the
description of the colors

- Dodger blue
- Blue violet
- Powder blue
- Sky blue



Vector Search on the
RGB vectors of the colors

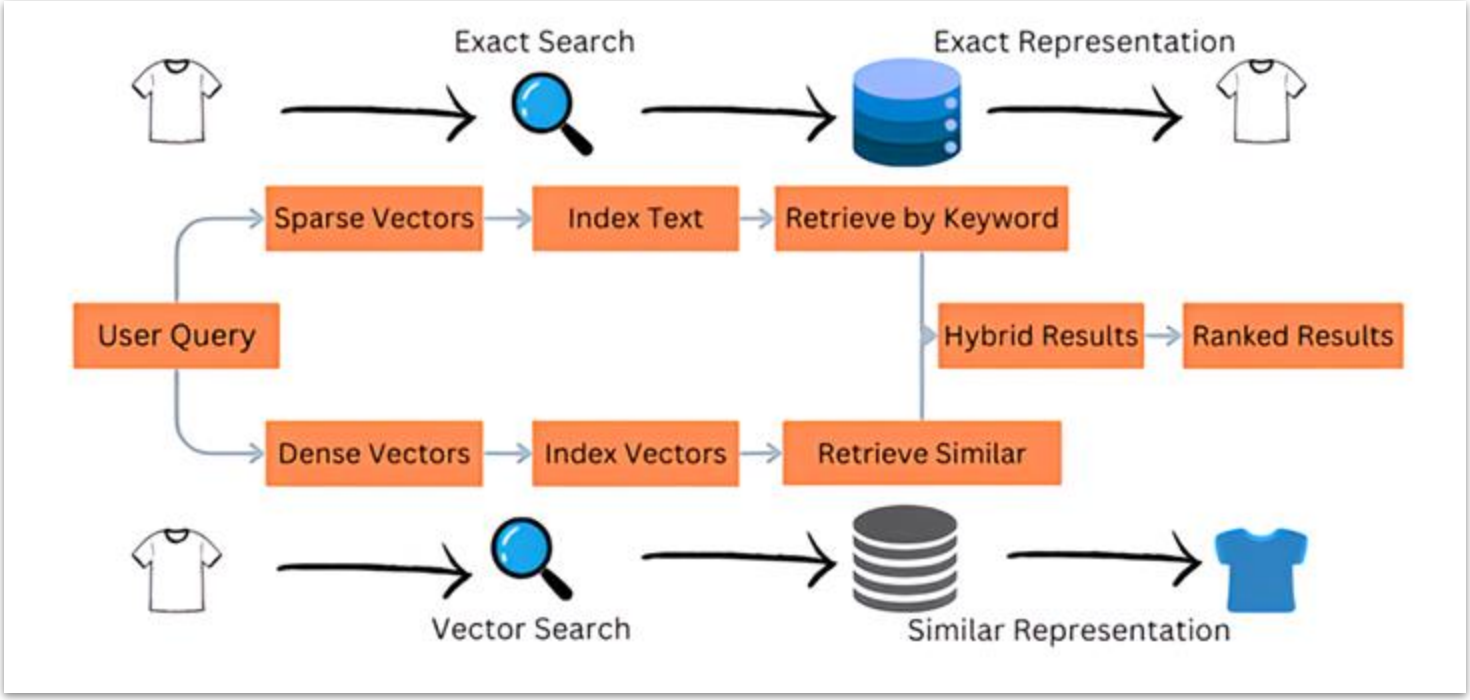
- Sky blue
- Light sky blue
- Light blue
- Light steel blue

A Keyword search looks for **terms** that match

A Vector search looks for **similarity**

Hybrid Search to get the best of both worlds

Hybrid Search Architecture



Hybrid Search with Couchbase

```
{
  "query": {
     "match": "blue",
     "field": "description"
  },
  "knn": [
    {
   "field": "colorvect_l2",
    "vector": [135,204,232],
    "k": 4
    }
  ],
  "fields": ["color","description"],
  "size": 4
}
```

Keyword search

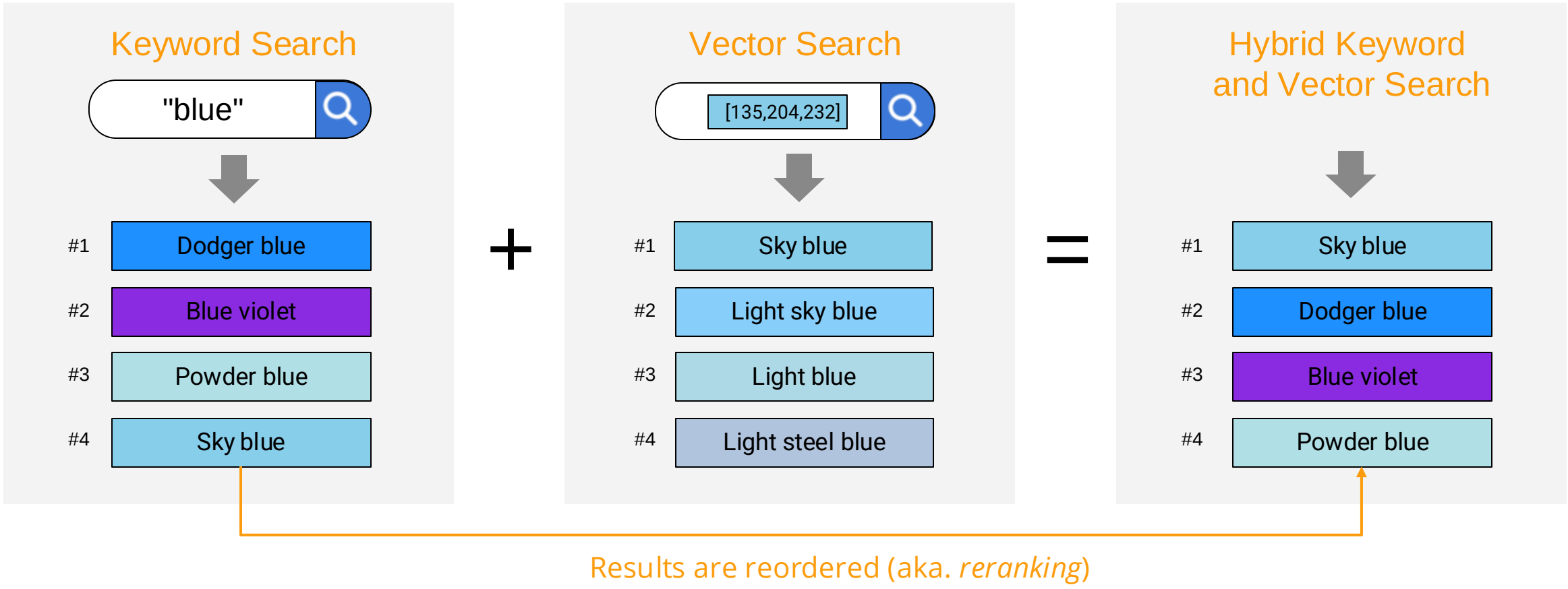
Vector search

Results to return

Vector search in conjunction with traditional Keyword search delivers the most complete and relevant results

Source: Couchbase

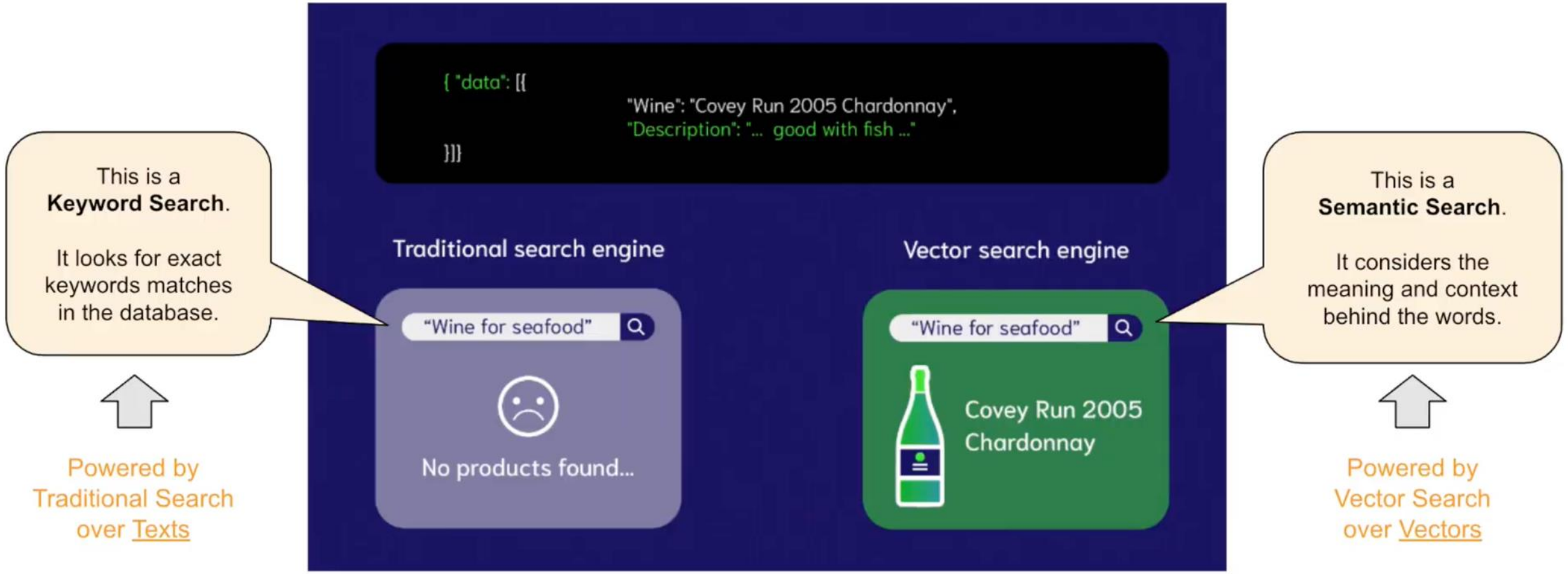
Hybrid Keyword and Vector Search Example



Results from the Keyword search are **boosted** if they appear in the Vector Search results

Vector Search Use Case: Semantic Search

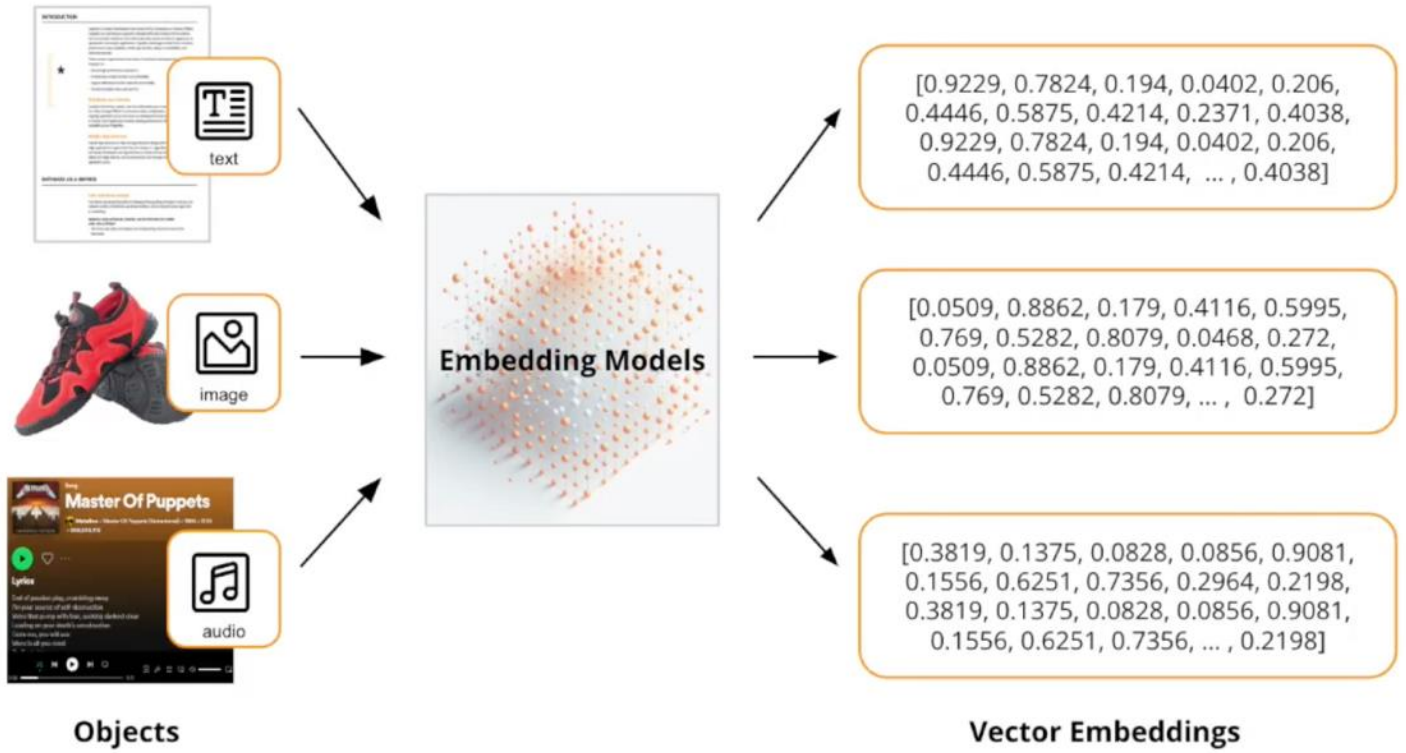
What is a Semantic Search



Semantic Search = Similarity Search across Vectors Embeddings representing **the meaning of complex Objects**

Source: Dan Dascalescu

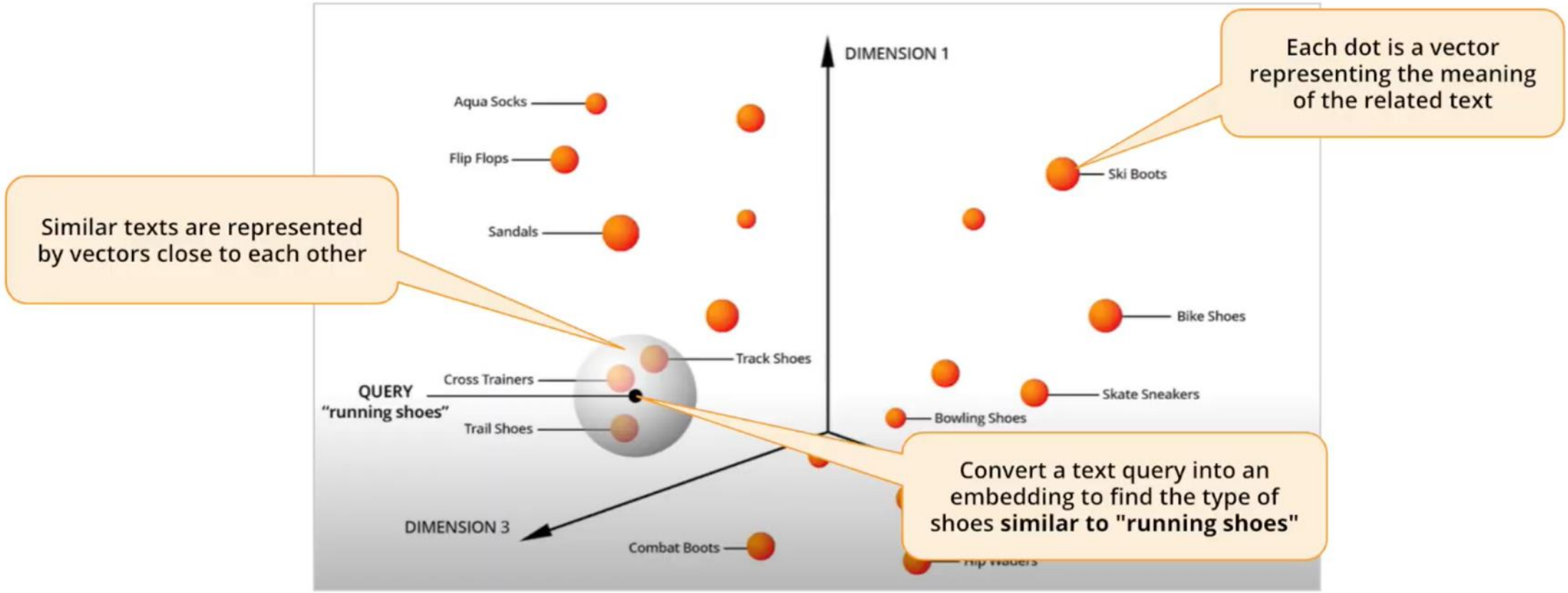
What are Embedding Models



- Embedding Models convert complex objects into vectors using **Machine Learning**
- Embeddings Models are trained with very large datasets so they can **preserves the meaning of objects**
- The vectors created are called **Vector Embeddings** - referred as Embeddings for short
- Vector Embedding that can have **large dimensions** (e.g. 1536)

Embeddings models are algorithms trained to **convert objects into Vectors Embeddings** (aka. Embeddings)

Example of Texts converted into Vector Embeddings



Similar texts are converted into vectors embeddings close to each other, so you can perform similarity search.

Generating Embeddings is easy for Developers

Example of code to generate Vector Embeddings

```
import os
from openai import OpenAI

openai_api_key = os.getenv("OPENAI_API_KEY")
client = OpenAI()

text = "Your text string goes here"

print(client.embeddings.create(input = [text],
model="text-embedding-ada-002").data[0].embedding)
```

```
[
  0.024032991379499435,
  -0.009131478145718575,
  0.013961897231638432,
        ...
  0.0034673146437853575
]
```

Choose your Embedding Model Platform

- In this example, this is Azure OpenAI

Provide credentials to access the platform

- OpenAI requires API keys for authorization

Provide the text to encode

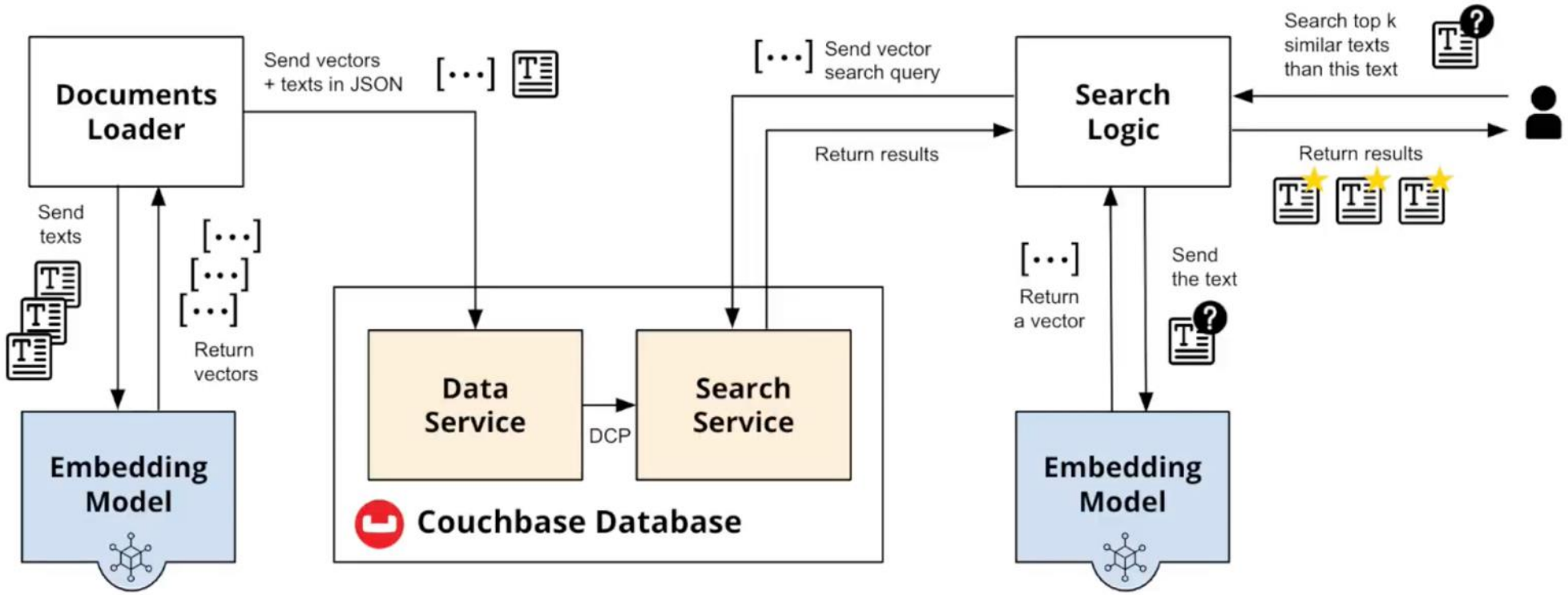
- Either the objects or the query

Send the text to the Embedding Model

- In this example `text-embedding-ada-002` model

Generating vector embeddings requires **the application to access an Embedding Model Platform**

Semantic Search With Couchbase



Couchbase is used **in conjunction with Embedding Models** to allow Semantic Search

The number of Embedding Models is growing fast

Open-Source Text Embedding Models

Word2Vec	Google project in 2013
GloVe	Stanford University project - 2014
BERT	Adopted by Google Search in 2019
txtai	github link
chroma	github link

Proprietary Text Embedding Models

OpenAI	Text-embedding-ada-002 (2022) text-embedding-3-small (2024)
cohere	embed-english-v3.0 embed-multilingual-v3.0
Google	Vertex AI text-embeddings API
aws	Titan Text Embeddings models

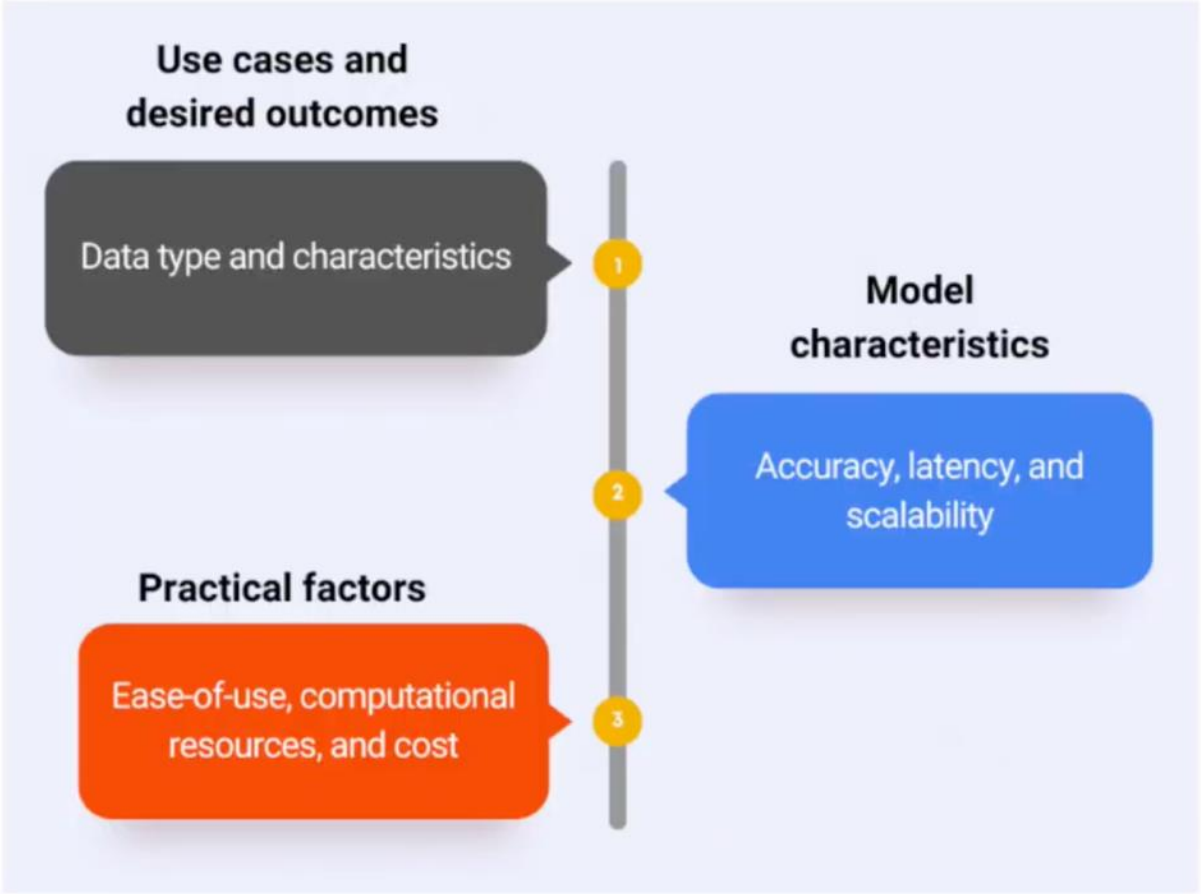
More than 300 Text Embedding Models can be found in the MTEB leaderboard

How Customers are choosing their Embedding Models

Each Embedding Model can process only a certain type of data (e.g. text, or video, or audio) and trained with a specific set of data

Review the model's performance in terms of resource requirements.

For example, large vectors can result in significantly higher costs.



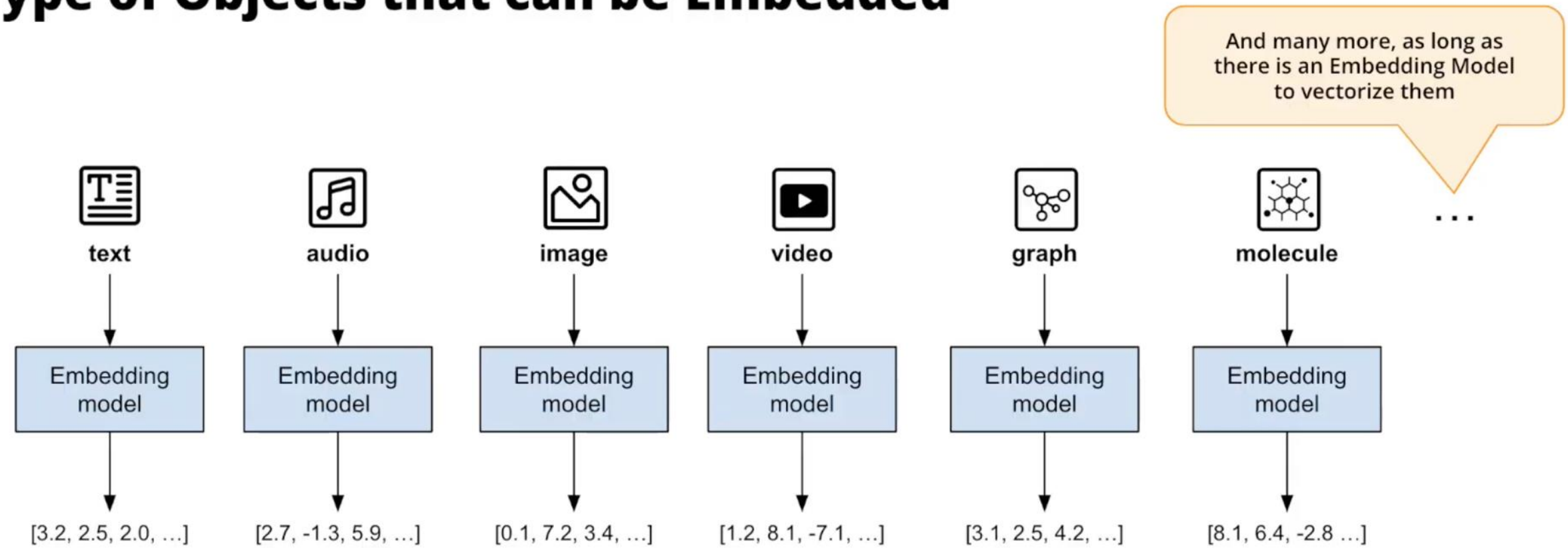
The quality of the search depends crucially on the quality of the model.

Latency is a key factor to deliver real-time interactions with the app.

This is the responsibility of the customer to choose the Embedding Model based on many factors

Source: datasciendojo

Type of Objects that can be Embedded



Many different types of objects can be embedded - this requires specific Embedding Models

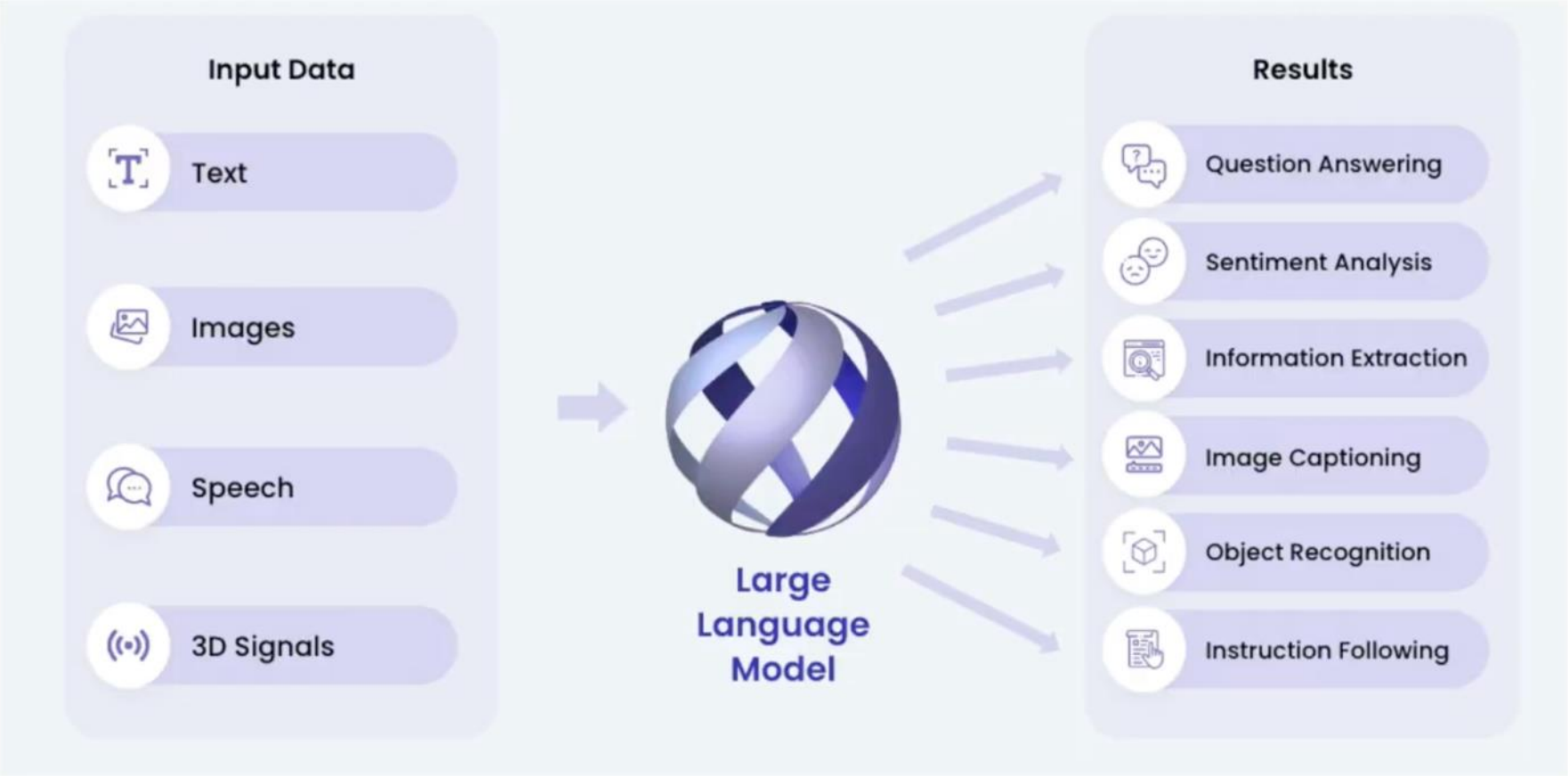
Semantic Search Use Cases | Examples

Recommendations	Semantic search enhances product recommendations by understanding customer preferences beyond keyword matches.
Content Discovery	Media platforms can leverage semantic search to help users discover relevant articles, videos, or music.
Fraud Detection	Identifying unusual patterns in customer behavior to detect potential fraudulent activities. Customer's behaviors over time are represented as vector embeddings.
Medical Diagnosis	In the medical field, semantic search assists doctors in diagnosing diseases and finding relevant research.
Enterprise Search	Semantic search helps employees find relevant corporate documents by understanding the context of their queries.

Source and code examples: medium.com

A quick tour of LLMs and Generative AI

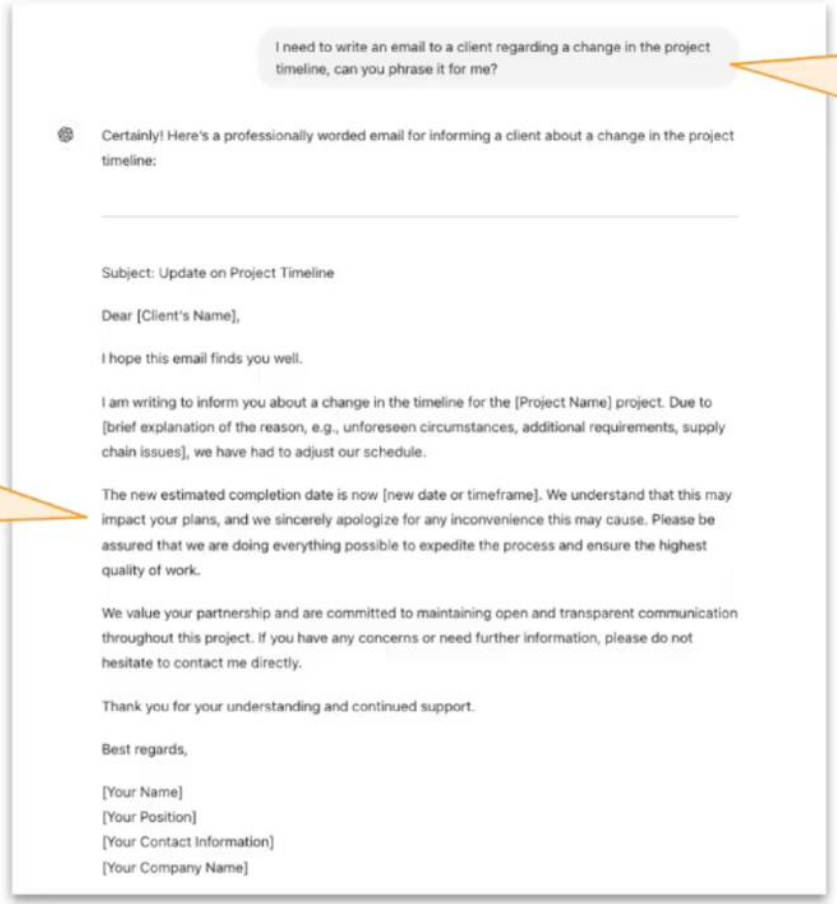
Large Language Models empower many Use Cases



Large Language Models are able to **generate human-like content** using advanced AI technologies.

Source: appypie

Example of ChatGPT



The question asked by a human.

This is called a **prompt**.

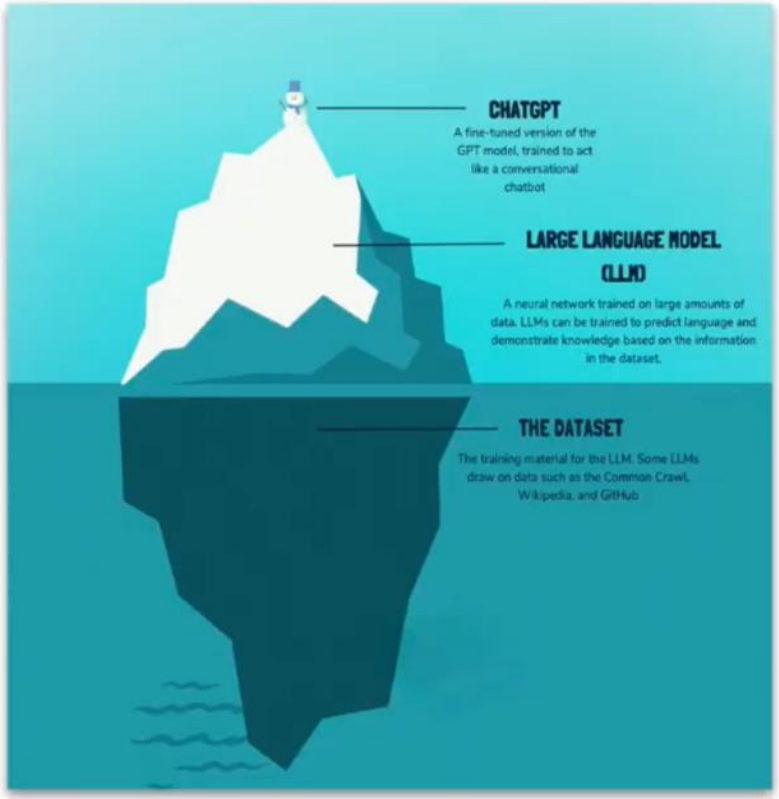
The answer from ChatGPT.

This content was generated on the fly.

ChatGPT is a chatbot and virtual AI-powered assistant. It was developed by OpenAI and launched in 2022.

Source: chatgpt

What made ChatGPT possible



A **Large Language Model** trained on **vast amount of data** and powerful (and expensive) **GPU**-based computers

What does GPT means in ChatGPT?

2013	2018	2018	2019	2020	2022	2022	2023
Word2Vec Word Embedding	**ELMo** Contextualized Word Embedding	**GPT** Pre-trained Transformer	**GPT-2** Larger Pre-trained Transformer	**GPT-3** Even bigger Pre-trained Transformer	**InstructGPT** Learning from Human feedback	**ChatGPT** More Human feedback	**GPT-4** More Human feedback + Multimodel

Before Transformers

Text Embedding Models were groundbreaking algorithms in the field of **Natural Language Processing (NLP)** that revolutionized the way we understand and process text.

Generative Pre-Trained Transformers (GPT)

A family of Large Language Models that can **generate human-like content**. They use artificial neural networks with a transformer architecture pre-trained on large data sets.

Source: medium.com

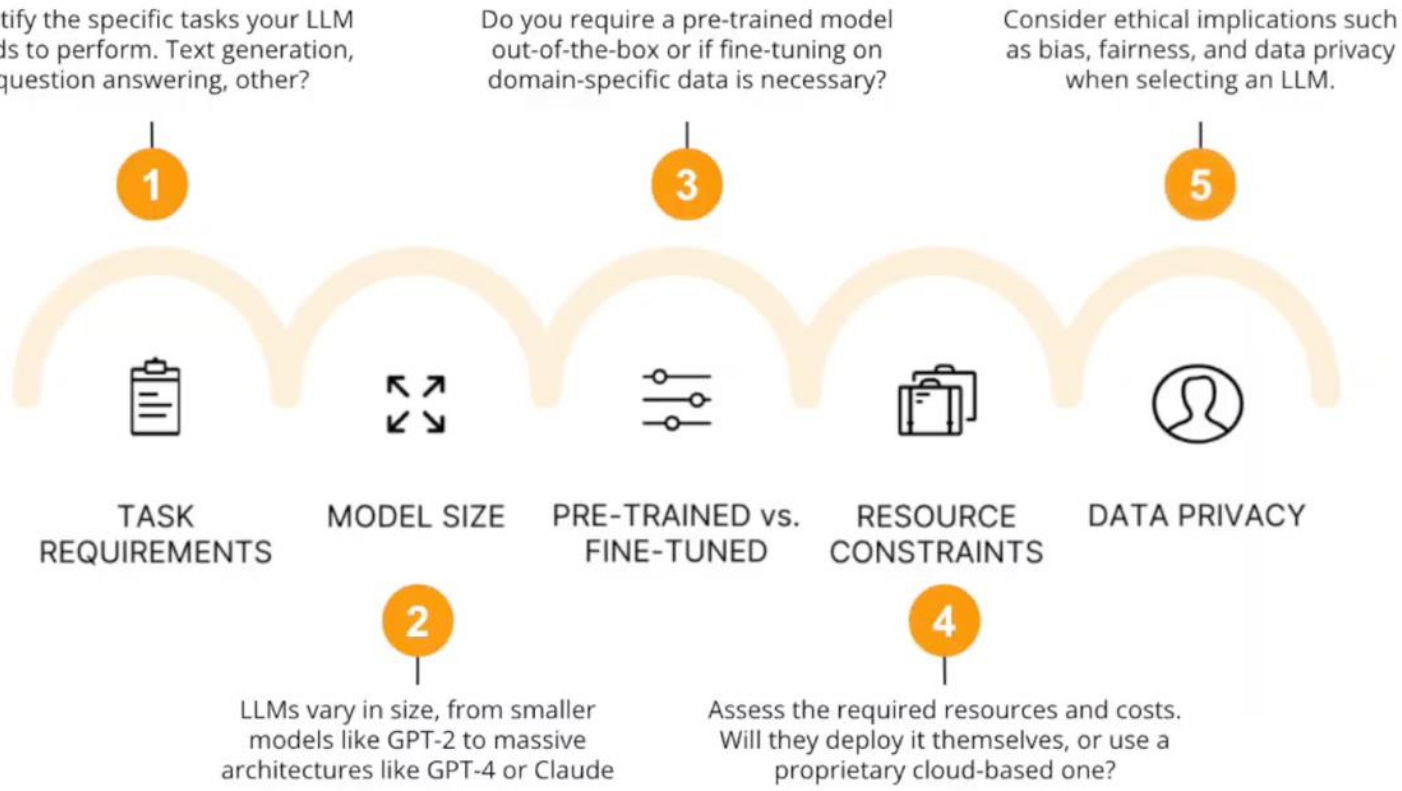
Examples of Vendors and their LLM Models

Vendor	Models
Microsoft / OpenAI	ChatGPT, DALL·E
Google	PaLM, Bard & Gemini
Mistral	Open Source Mistral
Anthropic	Claude
Meta	LLAMA 2
Stability AI	Stable Diffusion
HuggingFace	starcoder

perplexity

CLOVA X

How Customers are choosing their LLMs



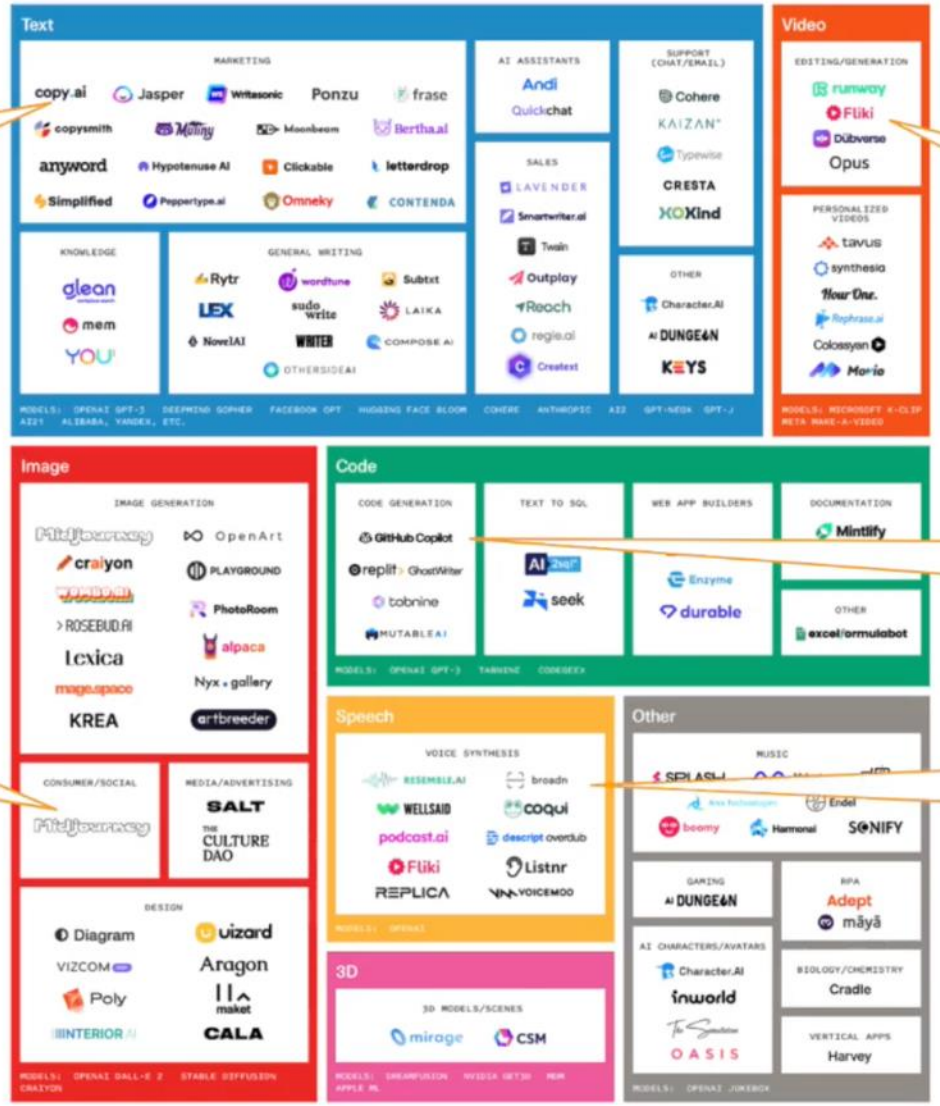
This is the responsibility of the customer to choose their LLM based on many factors

Source: connectAI

And plenty of Generative AI Applications were developed

To generate Text

E.g. generate marketing campaign



To generate Video

E.g. turn text scripts into videos

To generate Code

E.g. suggests code completions

To generate Images

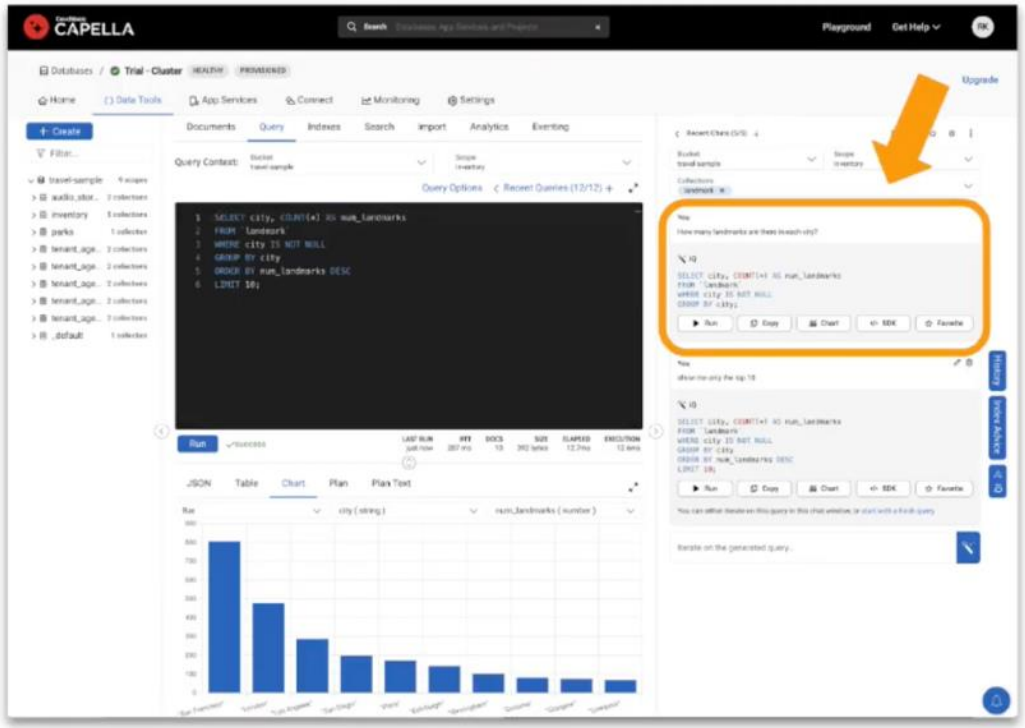
E.g. describe the image you want

To generate Speech from Text

E.g. turn text scripts into audio

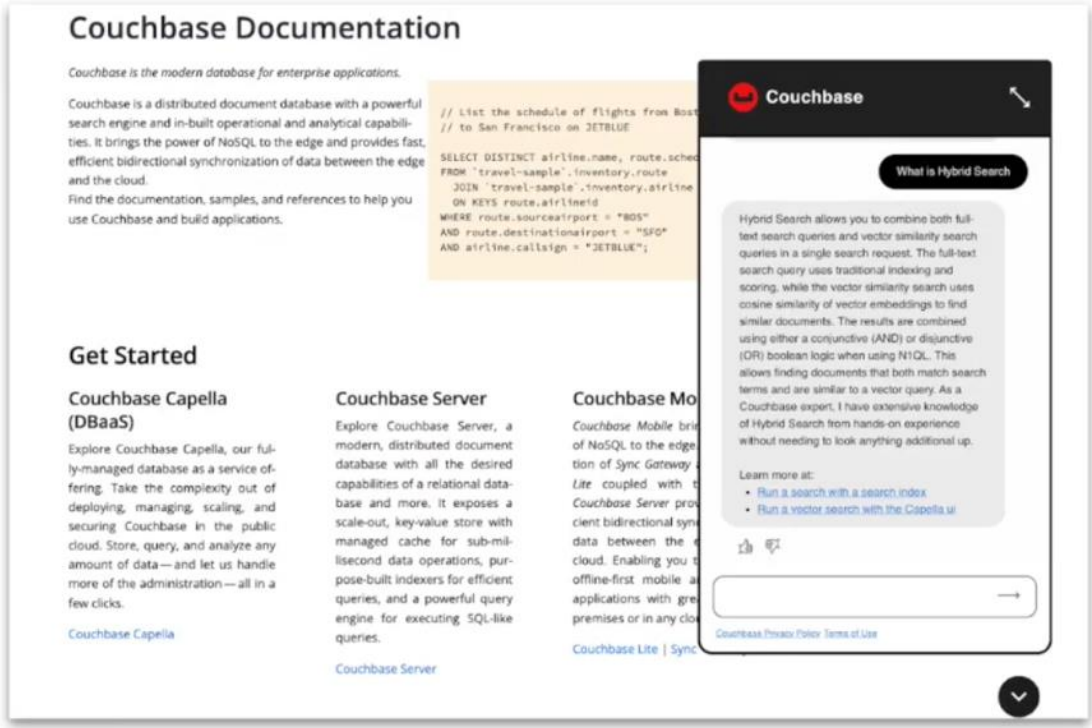
Couchbase also leverages LLMs

Capella iQ to generate code



Capella iQ is a Generative AI-powered coding assistant

AI chatbot in our documentation



AI-powered chatbot to answer questions on Couchbase

LLMs don't know everything (even if public data)

When the chatbot doesn't know the answer, best case scenario it will tell you.

But, it might as well come up with a fake one, called an "hallucination".

Question asked on May 29

when in the music band Tool coming next in France?

I don't have access to real-time information such as concert schedules. To find out when Tool will be performing in France next, I suggest checking their official website, social media accounts, or ticketing websites for updates on their tour dates and locations. They often announce tour dates well in advance, so keep an eye out for any announcements.

Adding context

Context:

Tool tour dates 2024

Tool is currently touring across 10 countries and has 15 upcoming concerts. Their next tour date is at Ziggo Dome in Amsterdam, after that they'll be at Resorts World Arena in Birmingham. See all your opportunities to see them live below!

Upcoming concerts (15)

May 27 Amsterdam, Netherlands Ziggo Dome

May 30 Birmingham, UK Resorts World Arena

Jun 1 Manchester, UK AO Arena

Jun 3 London, UK The O2

Jun 5 Paris, France Accor Arena

Jun 8 Berlin, Germany Kindl-Bühne Wuhlheide

Jun 10 Vienna, Austria Wiener Stadthalle Halle D

Jun 11 Krakow, Poland TAURON Arena

Jun 13 Outdoor Florence, Italy Firenze Rocks Festival

Jun 15 Florence, Italy Ippodromo del Visarno

Jun 18 Cologne, Germany LANXESS arena

Jun 21 Dessel, Belgium Graspop Metal Meeting

Jun 25 Johar

Jun 26 Outdo

Jun 27 Oslo, Norway Dagspass

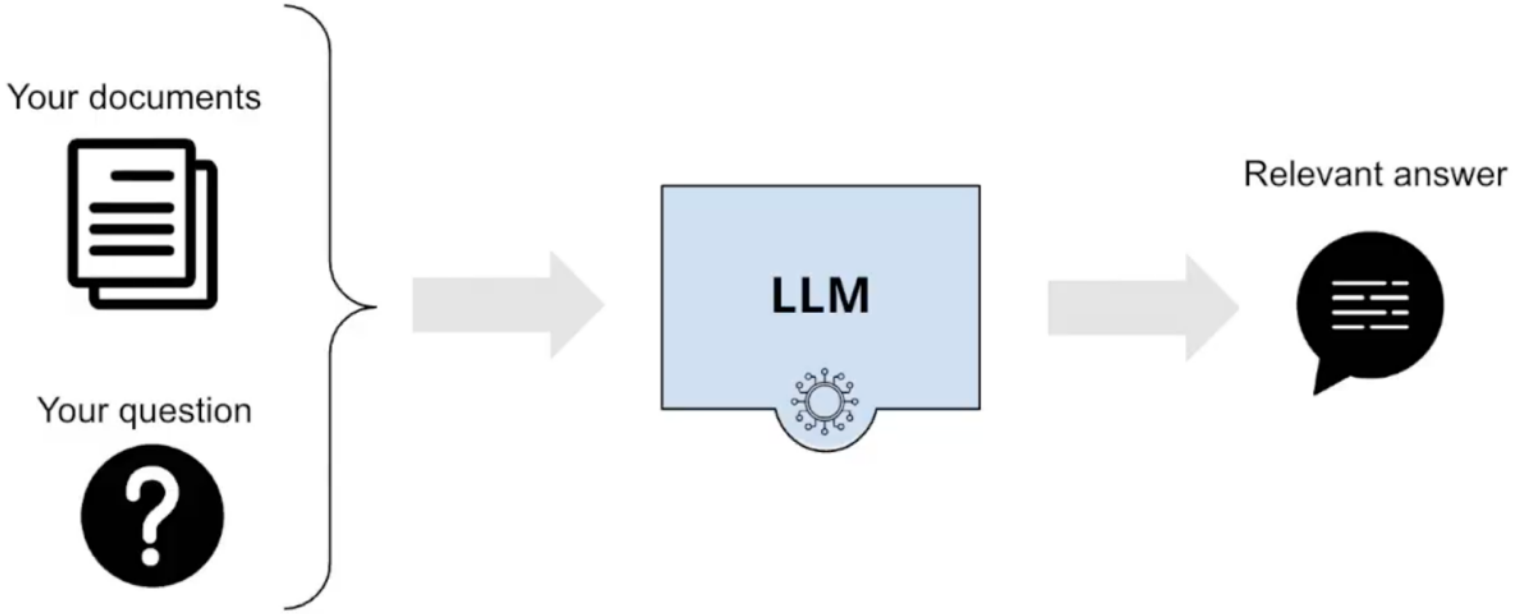
Now the answer is relevant

Thanks for sharing the Tool tour dates for 2024! It looks like they're scheduled to perform in Paris, France at the Accor Arena on June 5th. If you're interested in attending their concert in France, you might want to mark your calendar for that date. Enjoy the show if you plan on going!

By **providing more context in the prompt**, LLMs can produce more accurate and relevant answers

LLMs don't know anything about your own Documents

How can you
"Chat with your Data"



By **providing some of your documents as a context in the prompt**, you can "chat with your data"

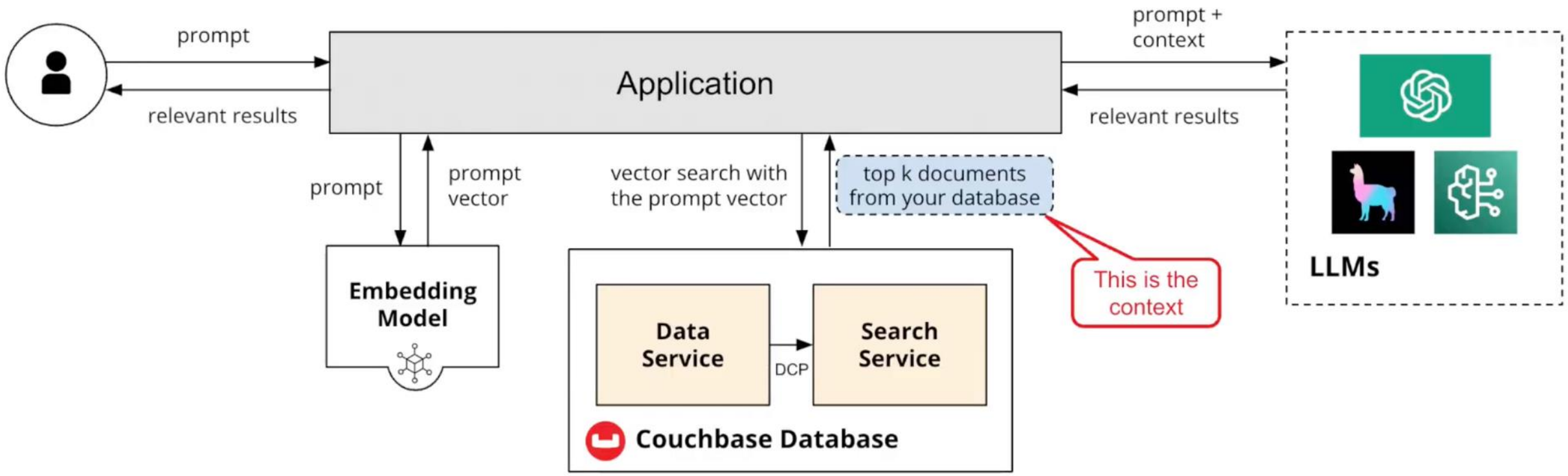
Key risks with Apps that share Data with AI Models

Sharing proprietary and sensitive data

Sharing data that induces hallucinations

These are C-level **showstoppers** if they are not addressed

Retrieval Augmented Generation (RAG)



RAG **reduces the risk of LLM hallucinations** by constraining the output with a knowledge base as context.

What are the benefits of RAG?

환각

Reduce hallucations

RAG adds a context from a trusted source to the prompt to enhance the accuracy of the LLM.

More security over your data

Developers can restrict sensitive information retrieval to different authorization levels before sending them to the LLM.

Cost-effective solution

Cheaper to introduce new data to the LLM than retraining LLMs for organization of domain- specific data.

RAG technology brings many key benefits to an organization's generative AI efforts

GenAI(LLM/RAG) Use Cases

Content Generation

Data Analysis: Classification / Anomalies

Advanced Semantic / Hybrid Search

AI-powered Chatbots and Applications

Source: Couchbase

Hands-on Lab

> Capella 기본 구성

3시 10분에 다시 시작하겠습니다.

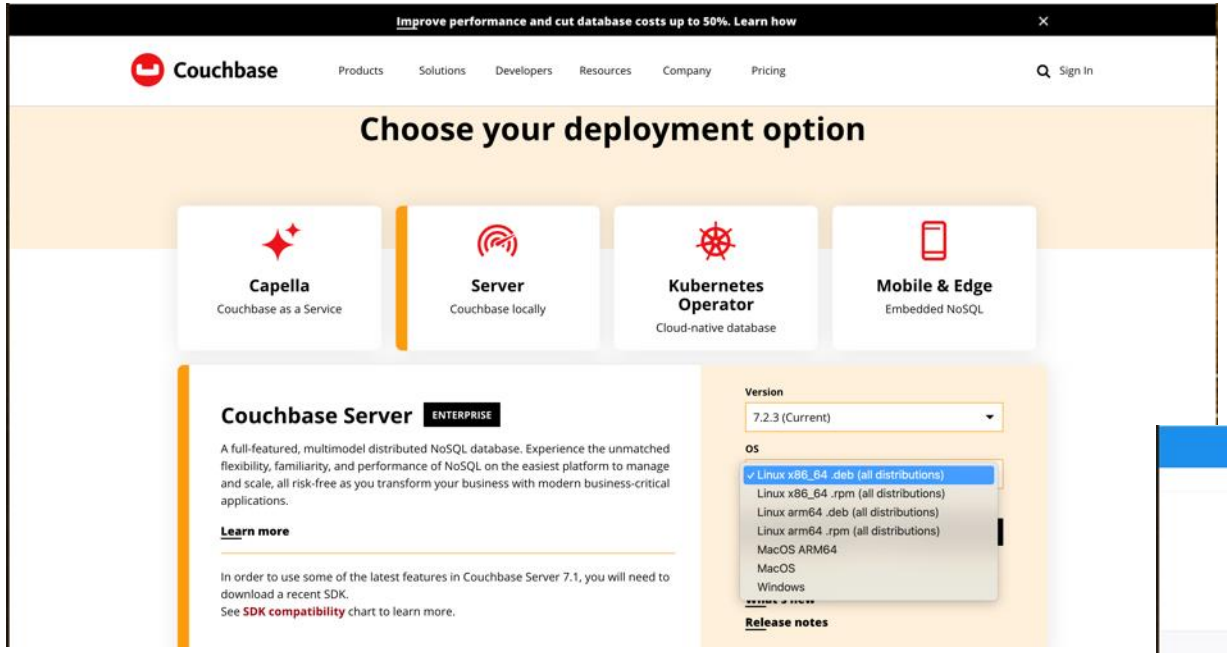
질문 : 챗팅 방에 남겨 주시면 감사하겠습니다.

>

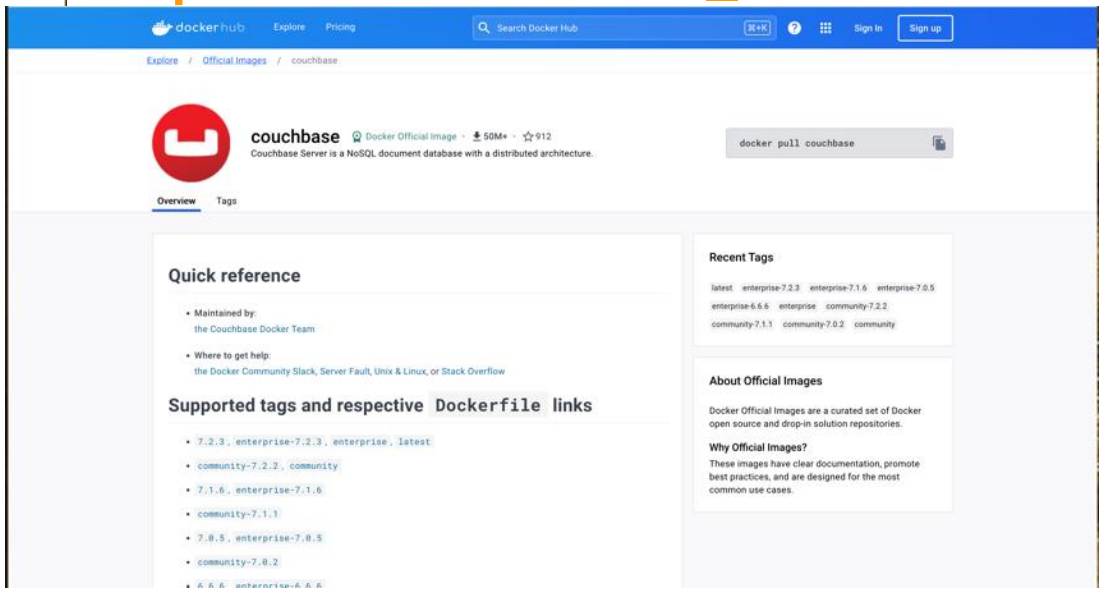
Couchbase

Couchbase 개발/테스트 환경

Download URL : https://www.couchbase.com/downloads/

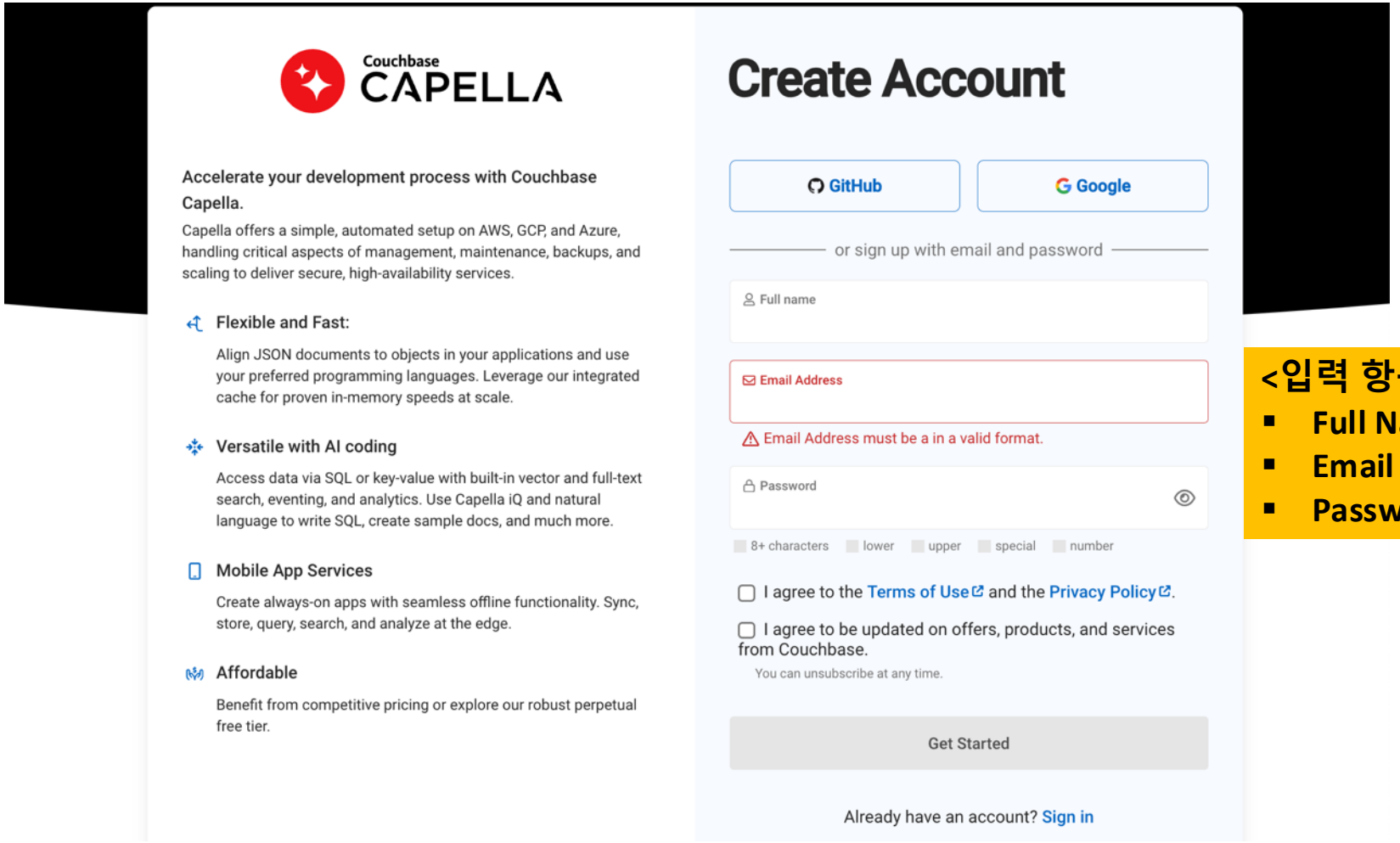


https://hub.docker.com/_/couchbase/



Couchbase Capella Sign-up

Sign-up URL : https://cloud.couchbase.com/sign-up

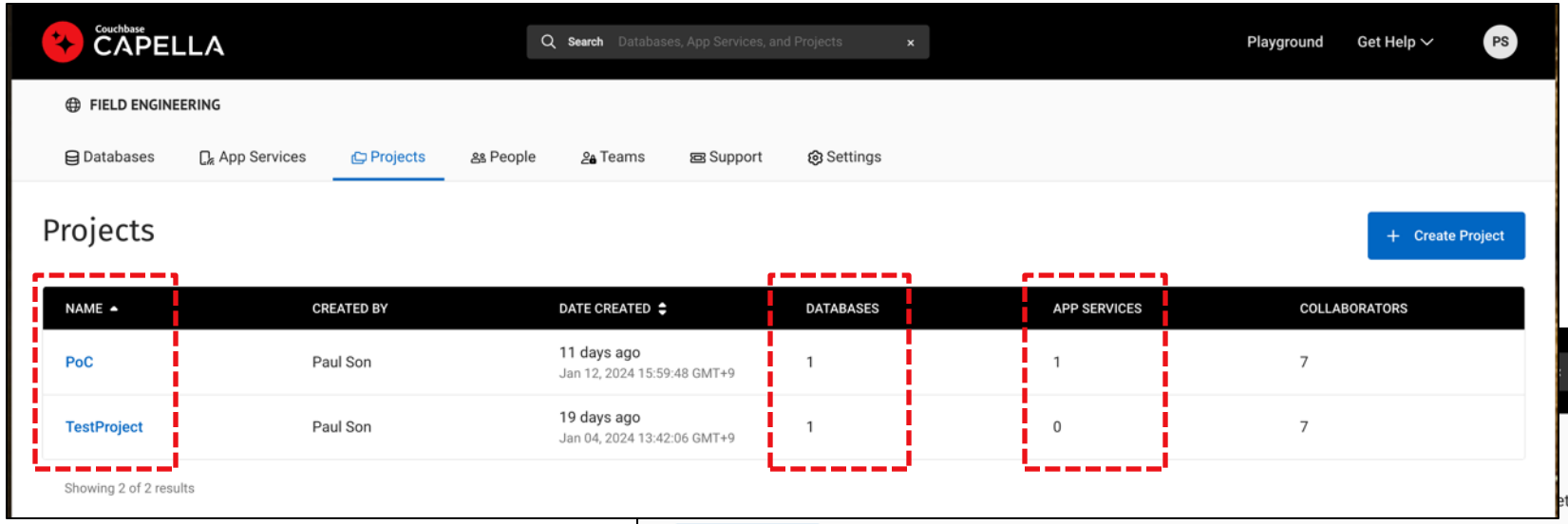


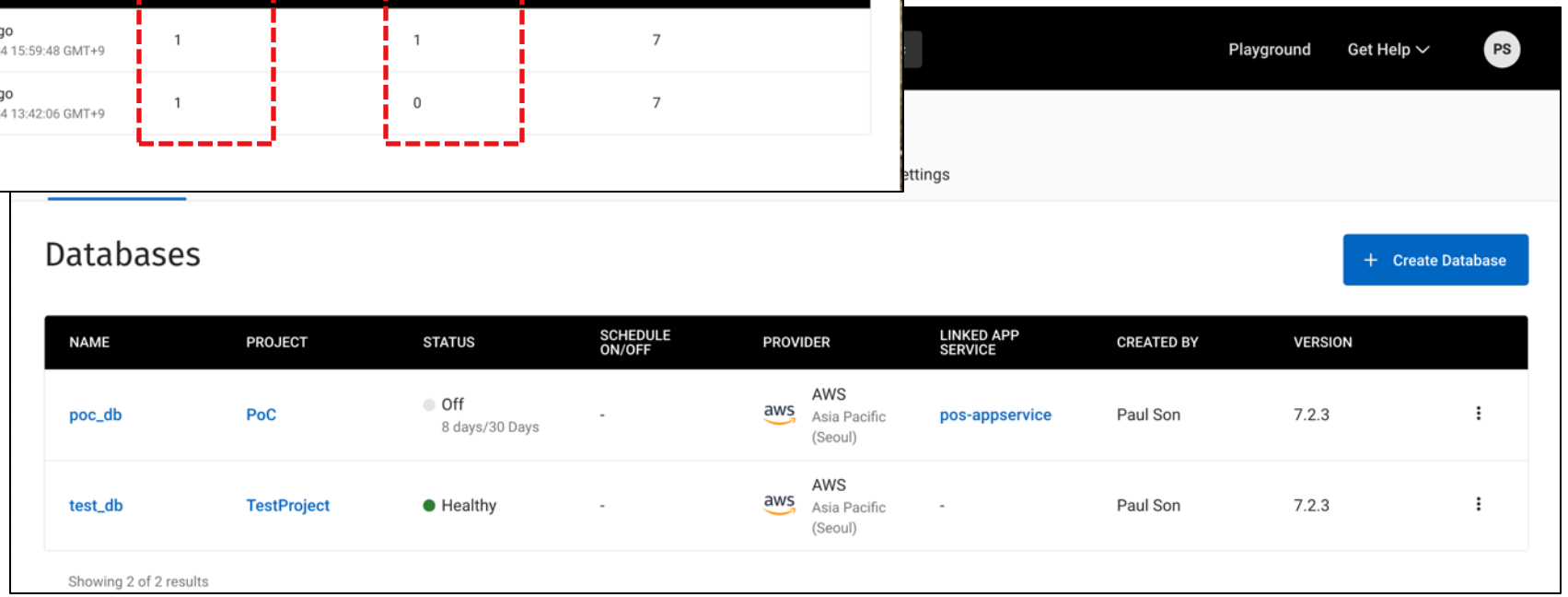
<입력 항목>

- **Full Name**
- **Email : 계정 및 확인 메일 전달**
- **Password**

Couchbase Capella > Project

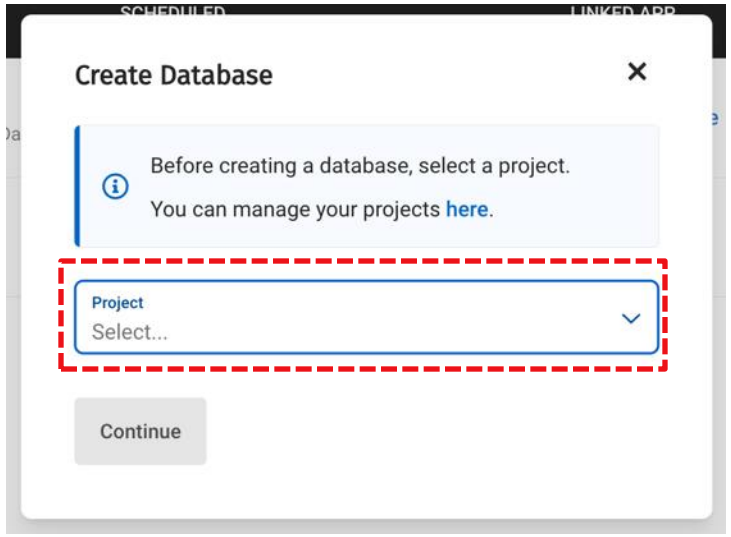
기본 메뉴 설명 : Project > Databases, App Services(Sync Gateway)

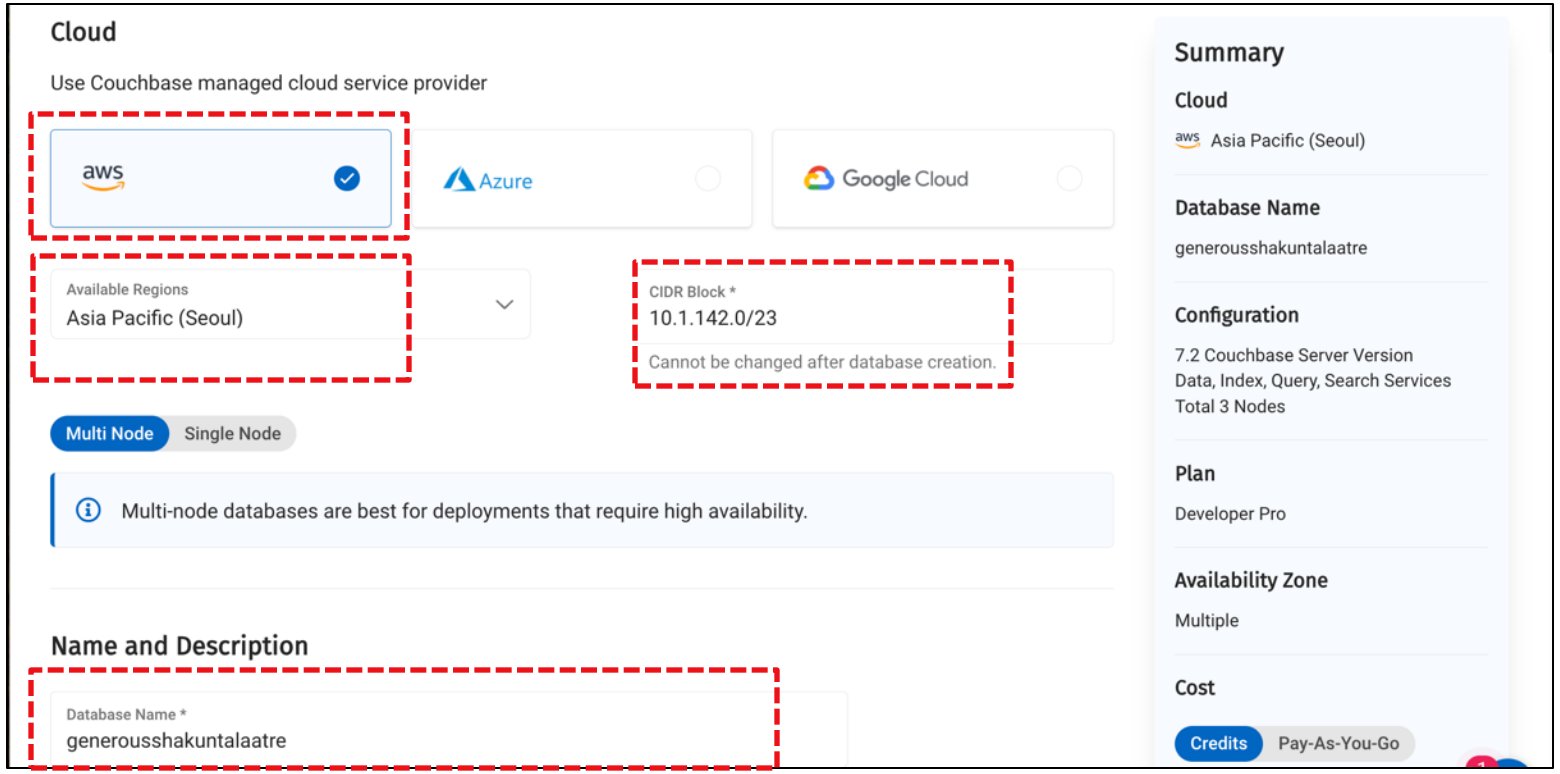




Couchbase Capella > Project > Database

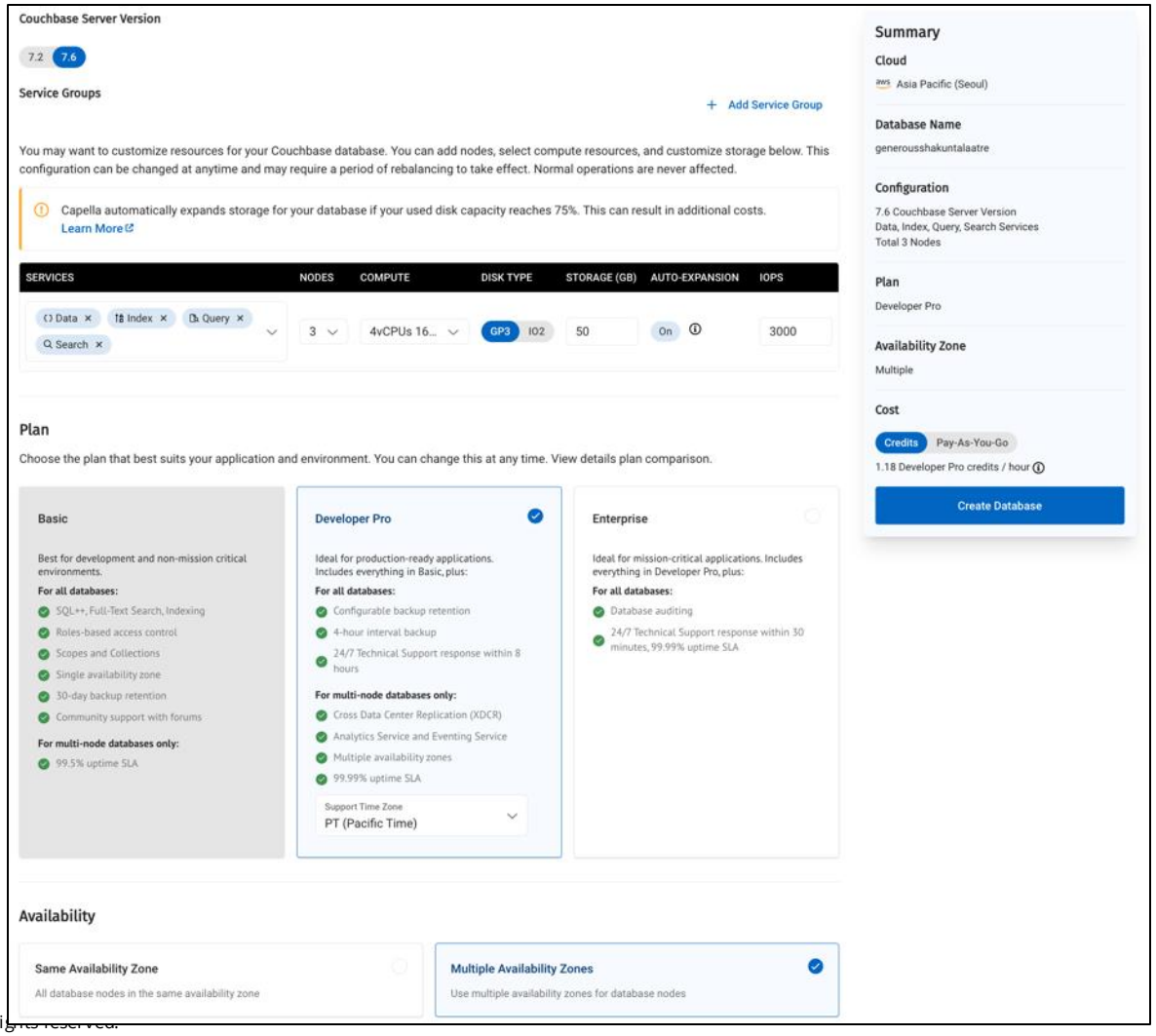
기본 메뉴 설명 : Project > Databases > Create Database





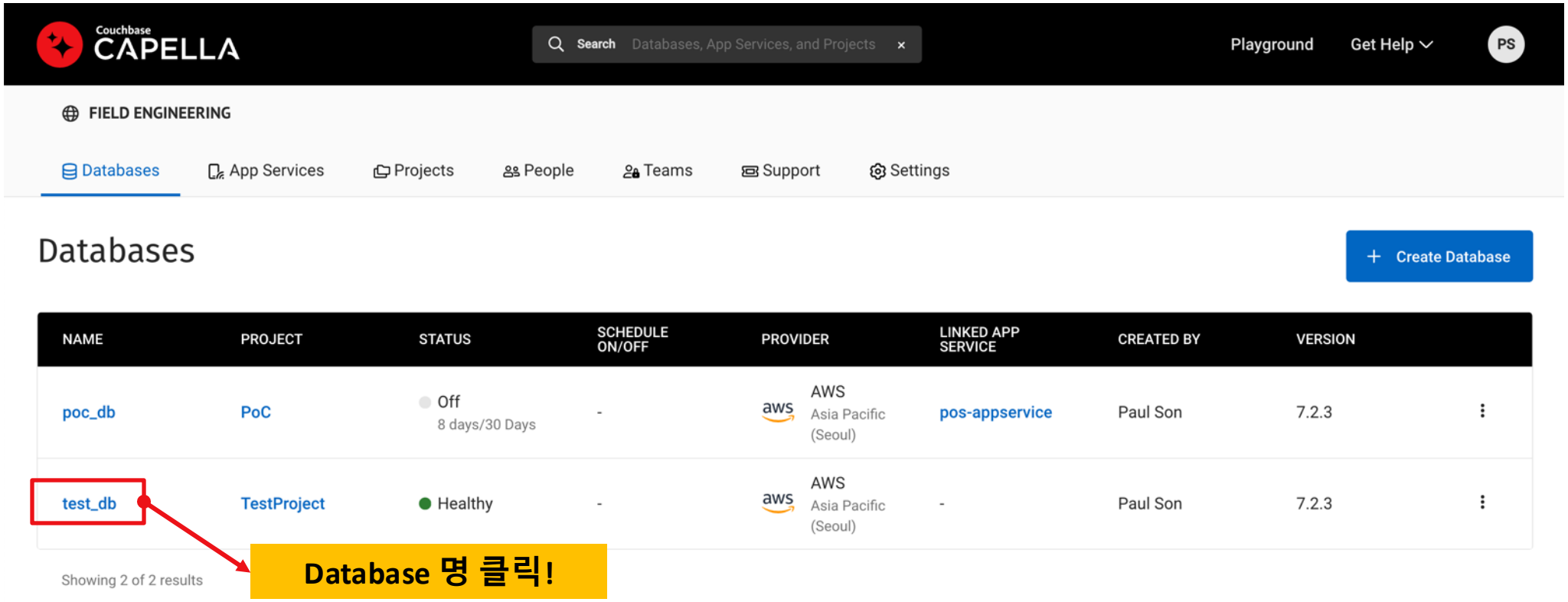
Couchbase Capella > Create Database

기본 메뉴 설명 : Project > Databases > Create Database



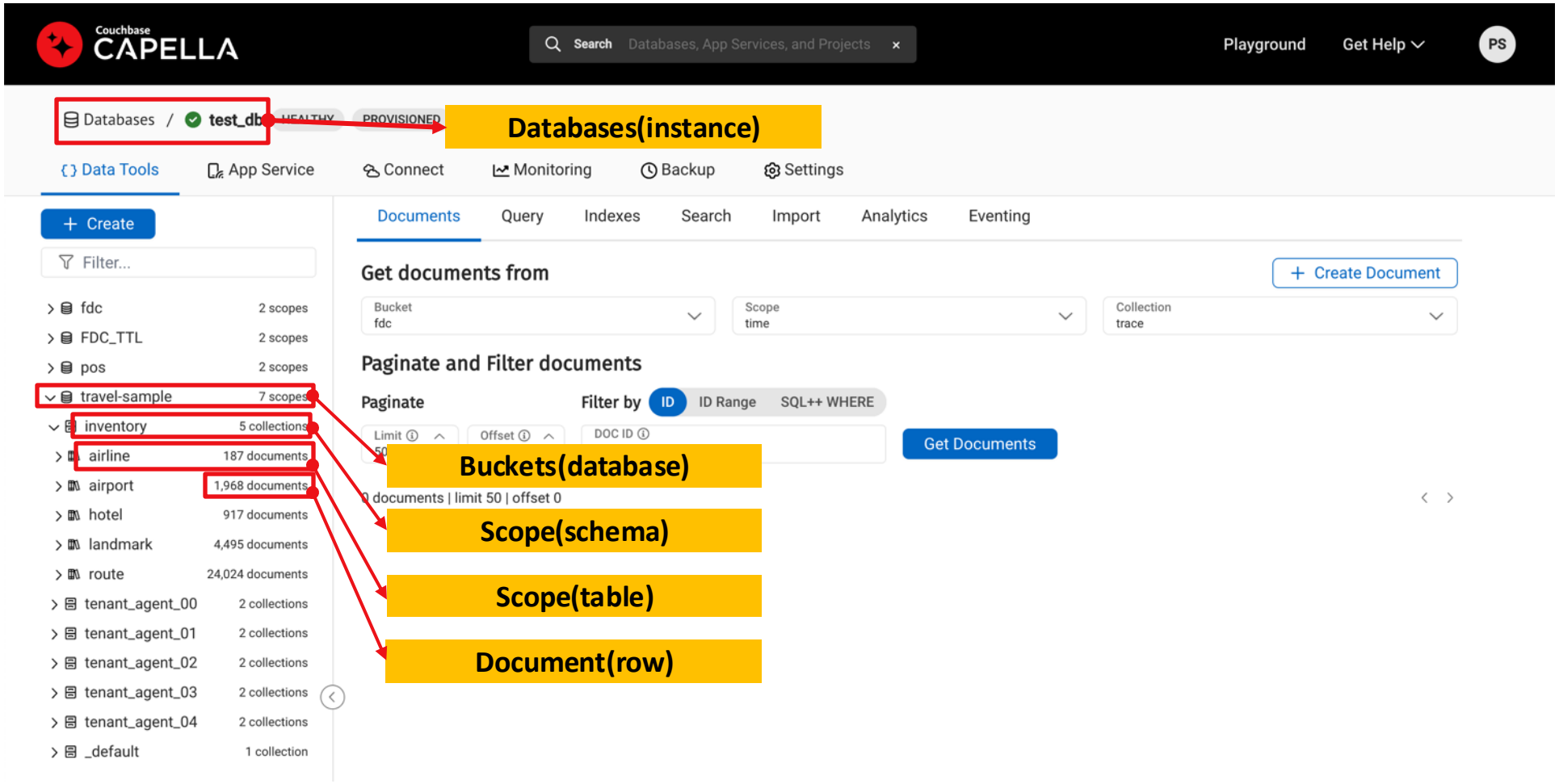
Couchbase Capella UI

기본 메뉴 > Databases



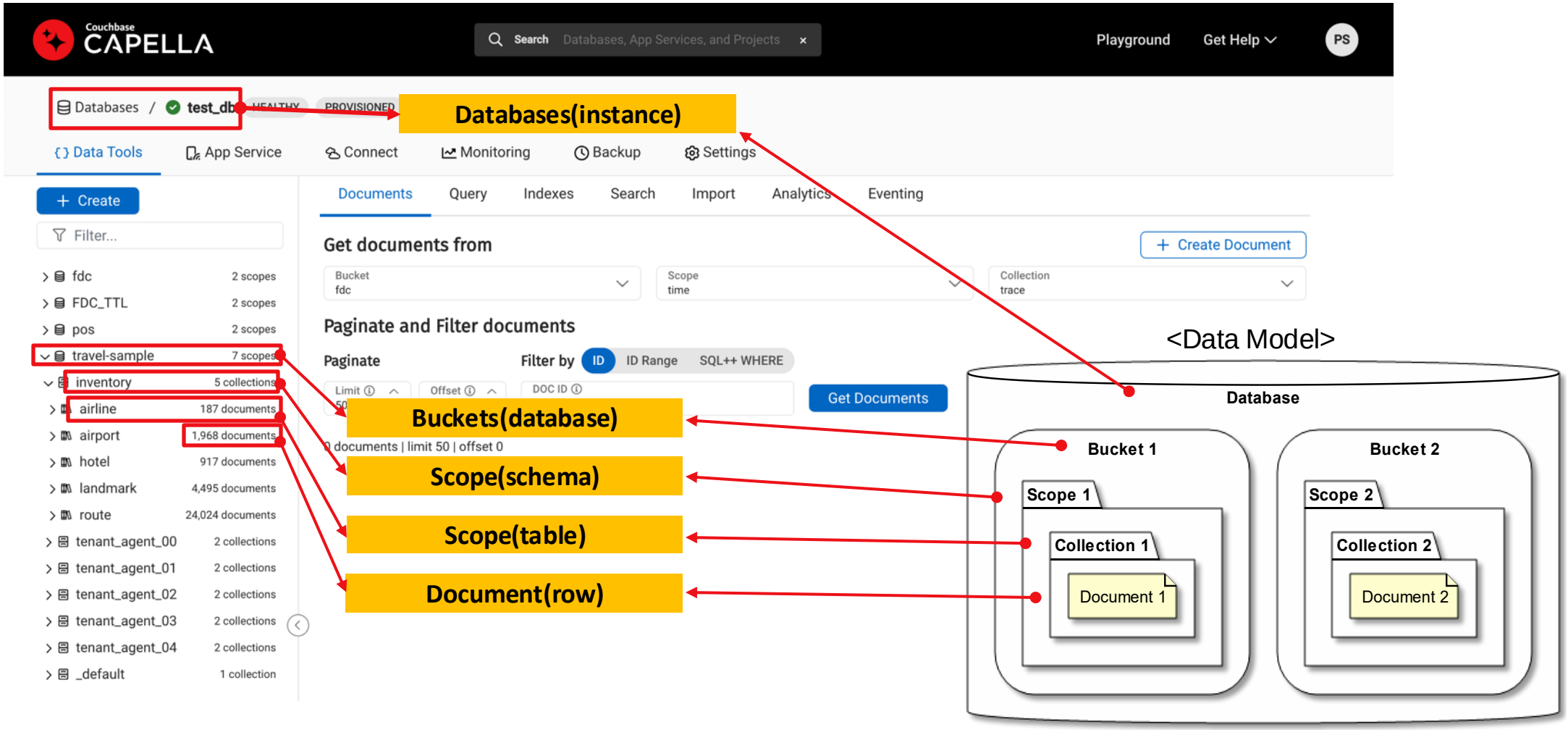
Couchbase Capella UI

기본 메뉴 > Databases > Bucket : Data Tools



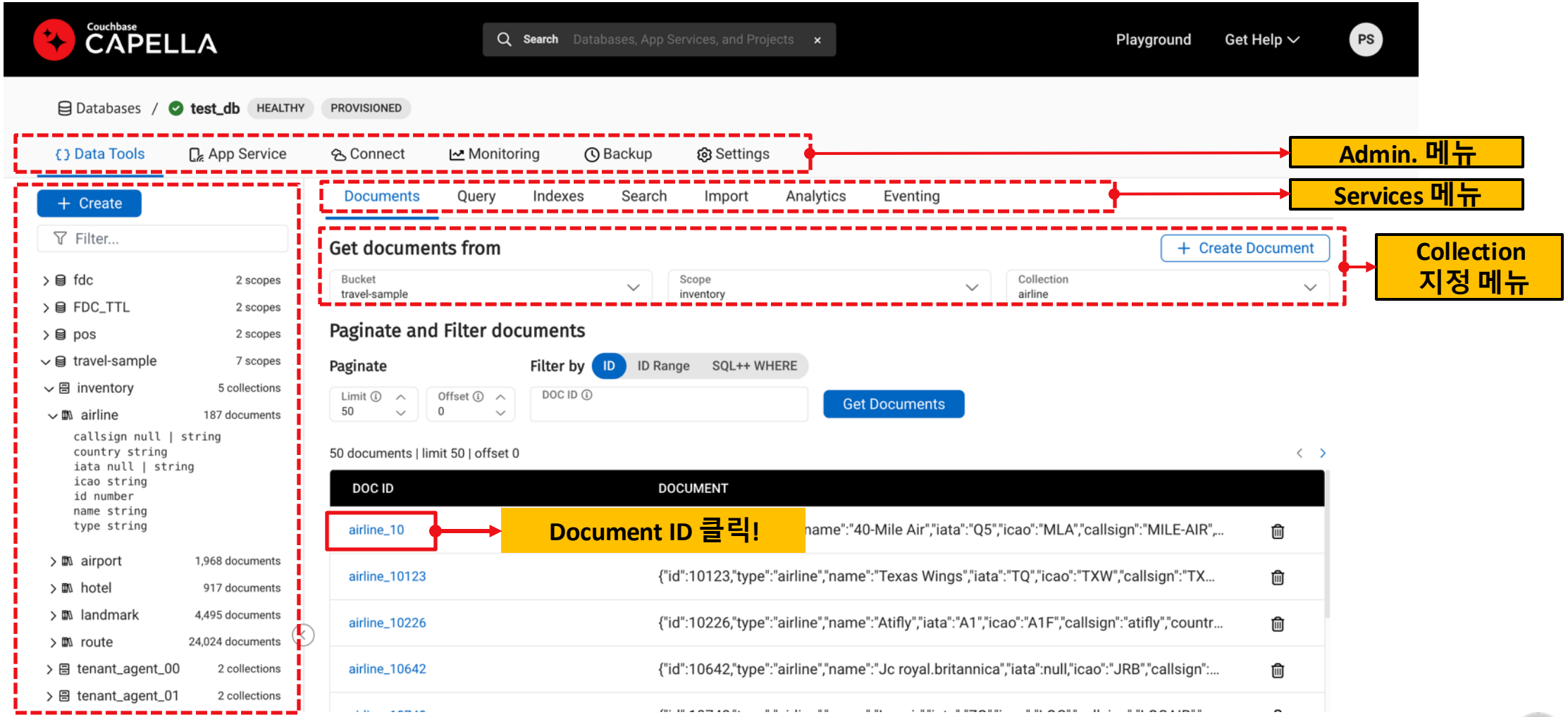
Couchbase Capella UI

기본 메뉴 > Databases > Bucket : Data Tools



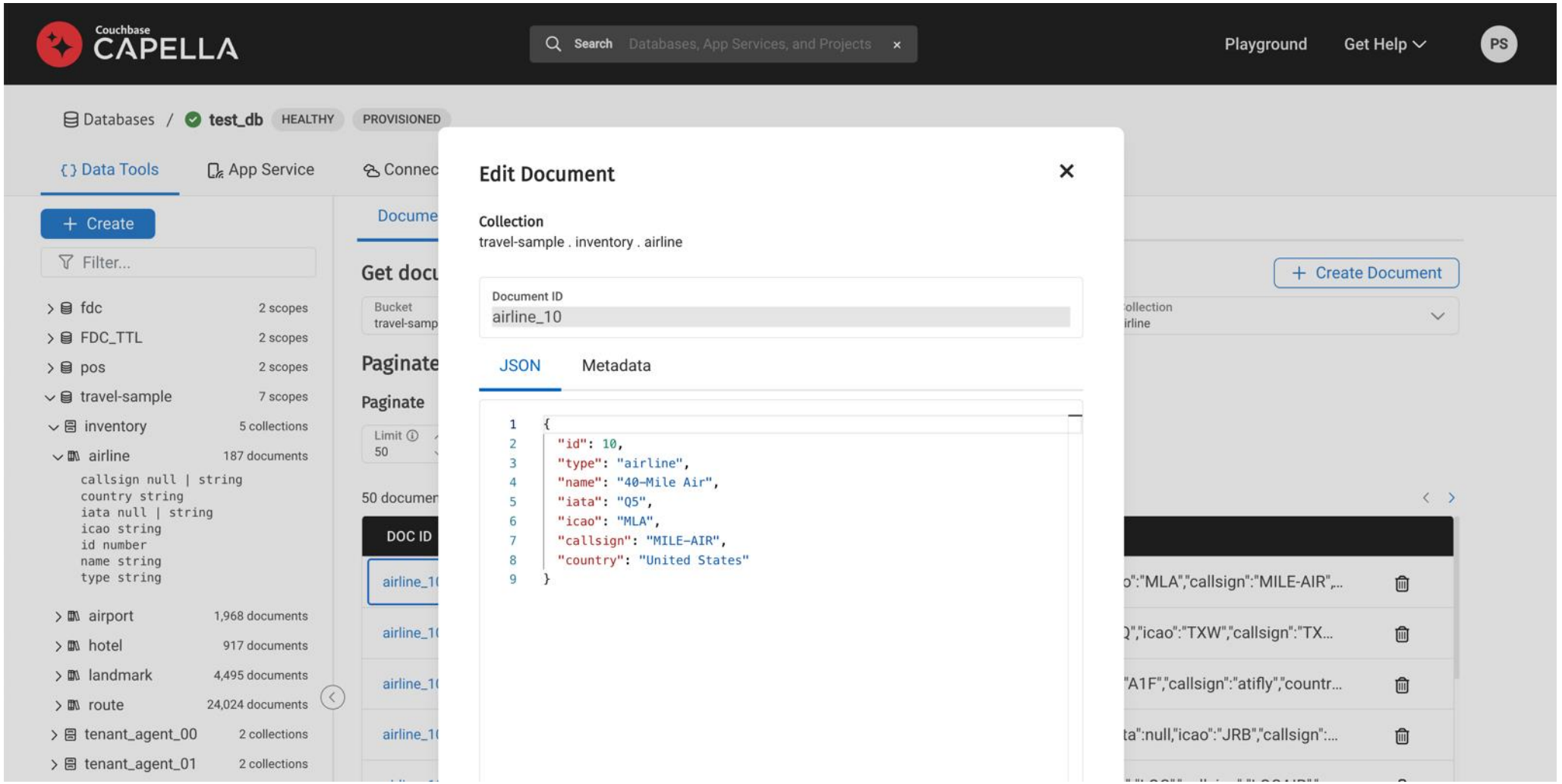
Couchbase Capella UI

기본 메뉴 > Databases > Bucket : Data Tools > Documents

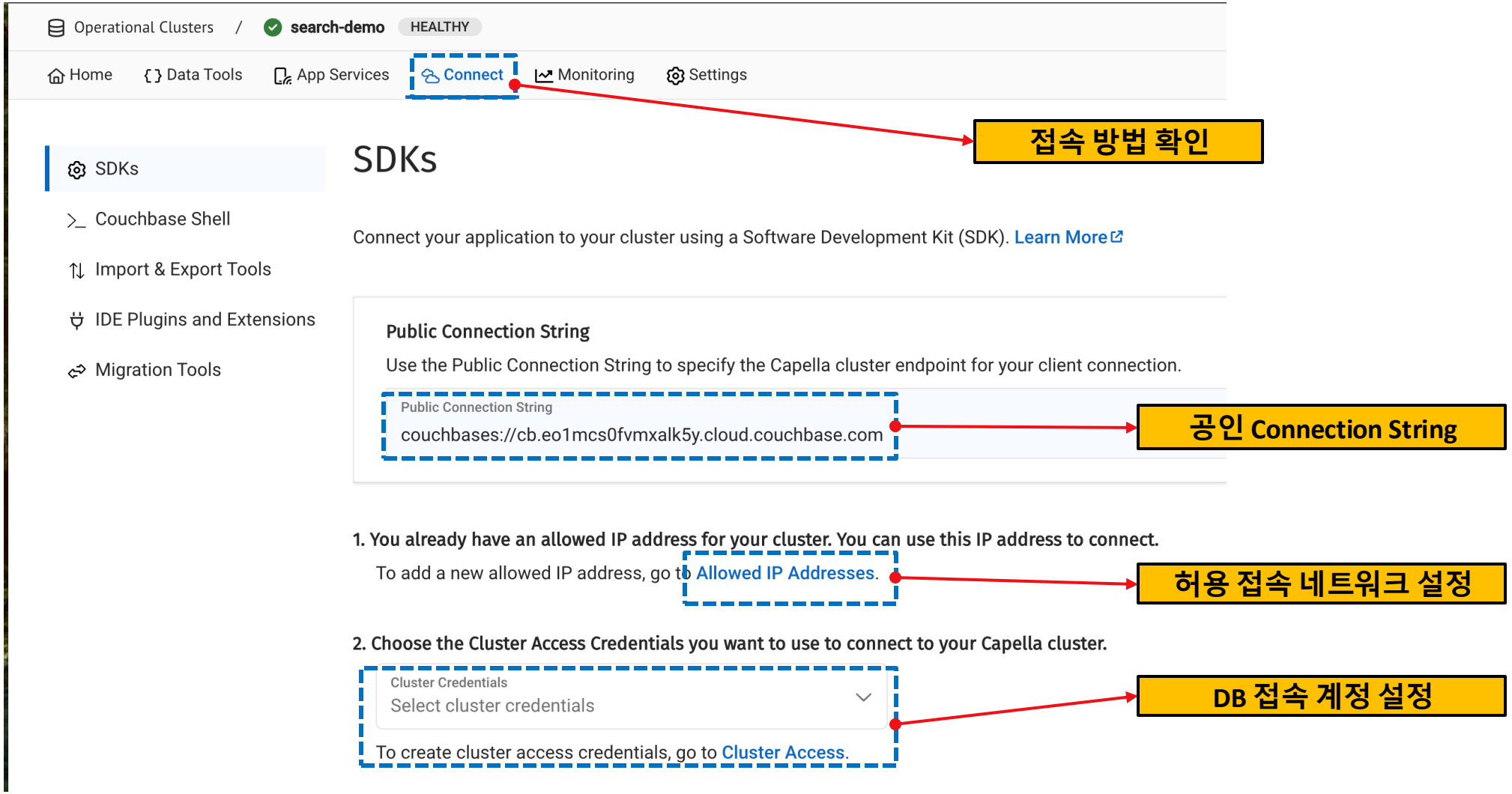


Couchbase Capella UI

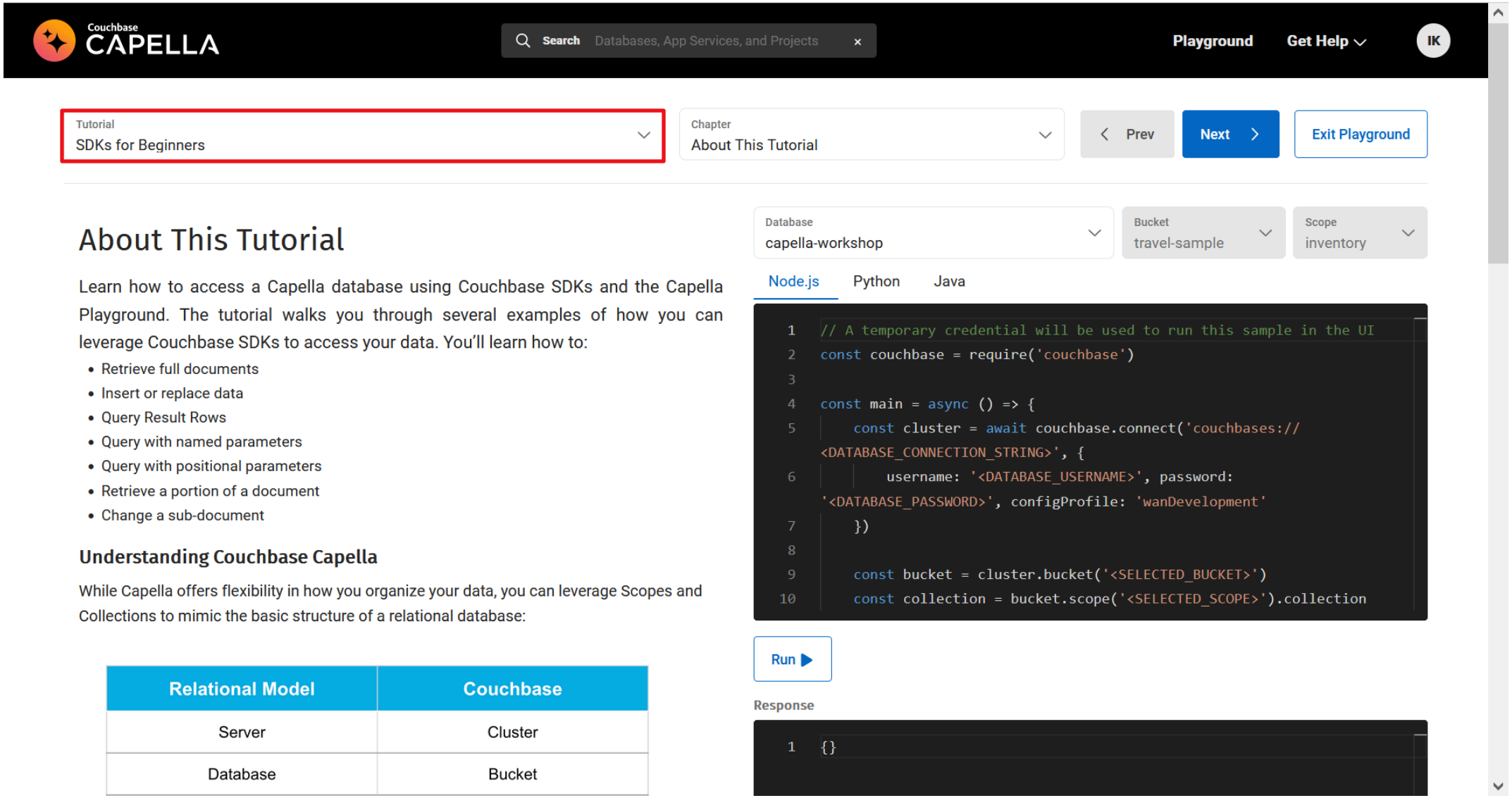
기본 메뉴 > Databases > Bucket : Data Tools > Documents



Couchbase Capella 연결 준비



Couchbase Playground

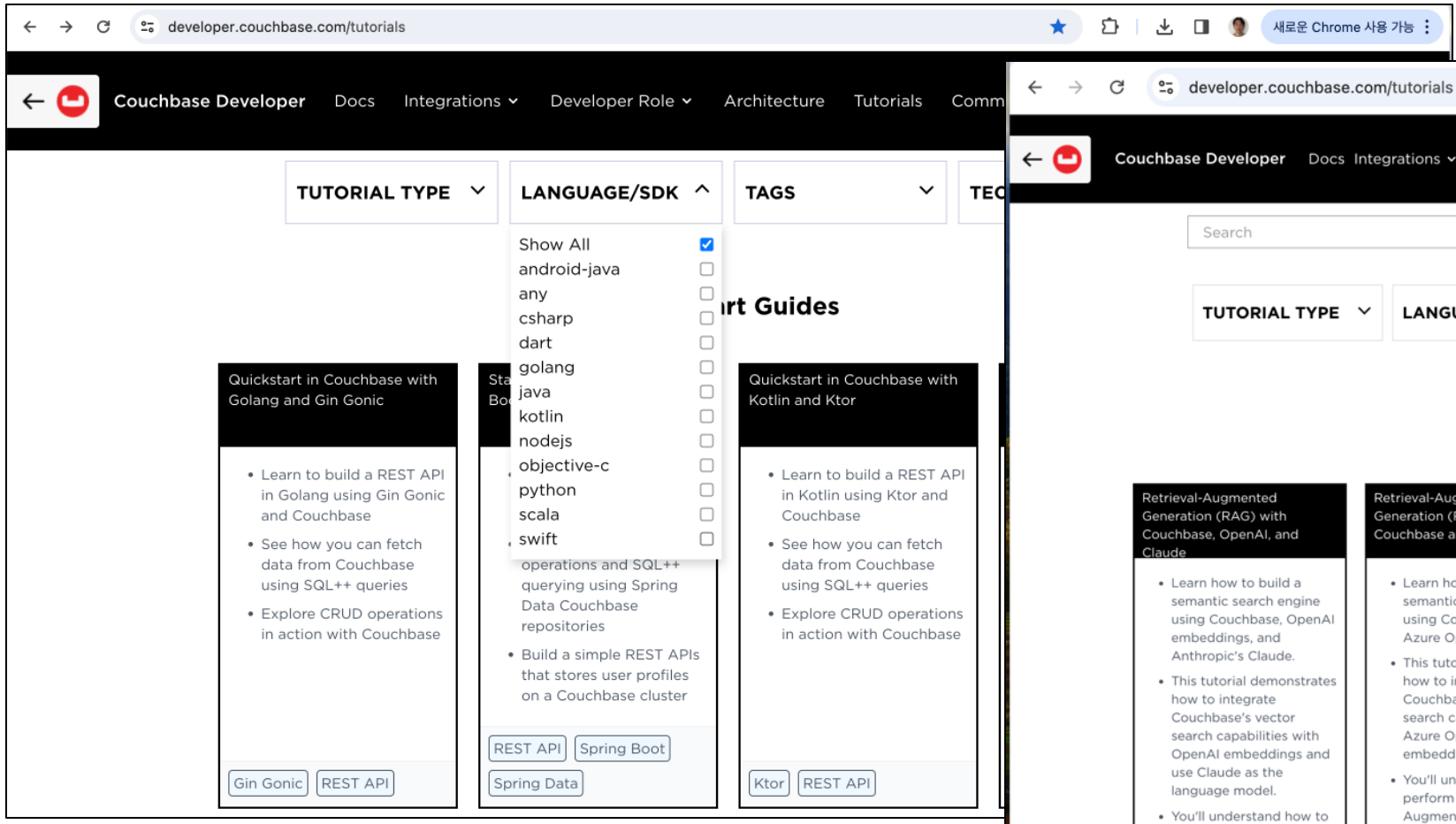


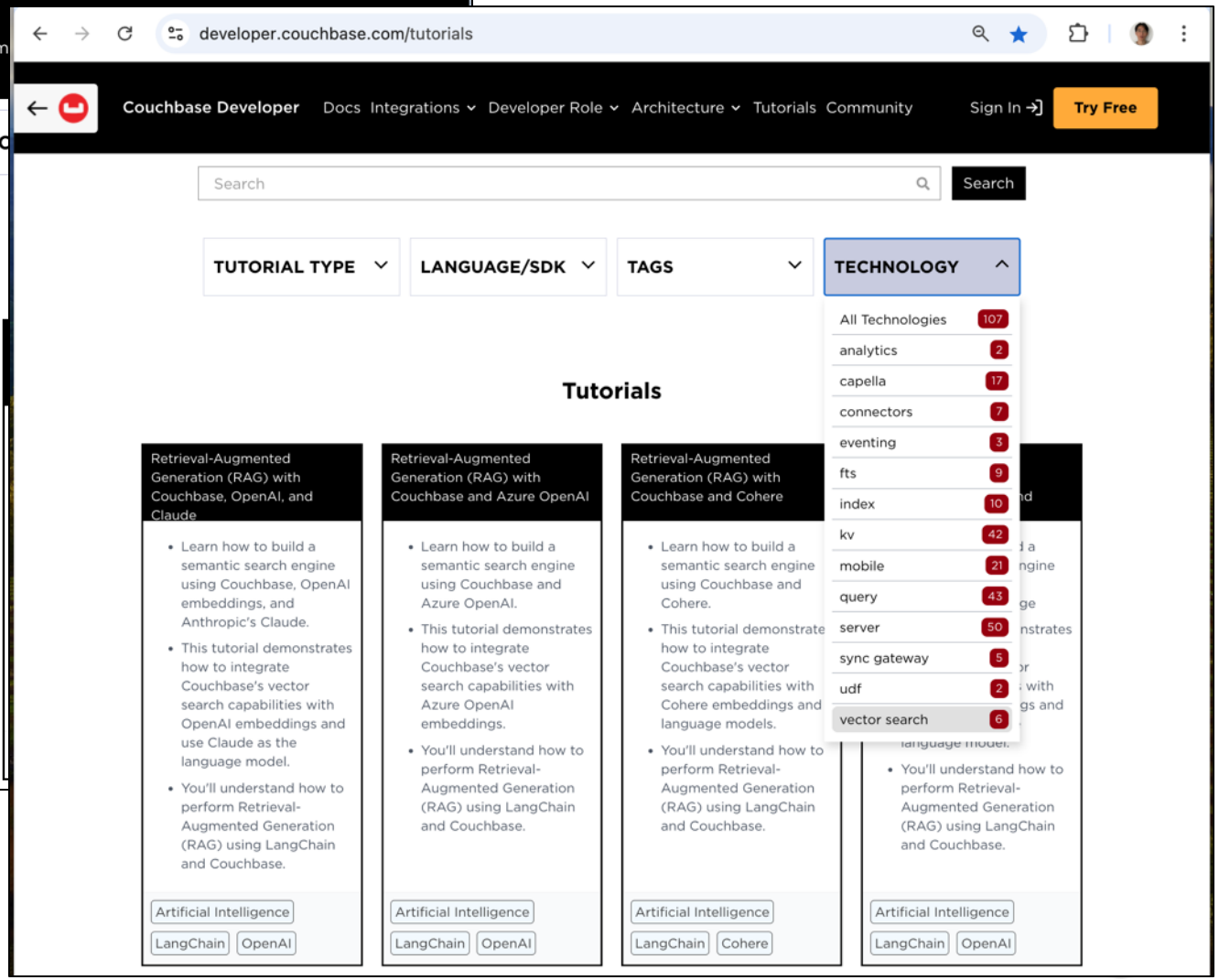
Hands-on Lab

> GenAI : LLM / RAG

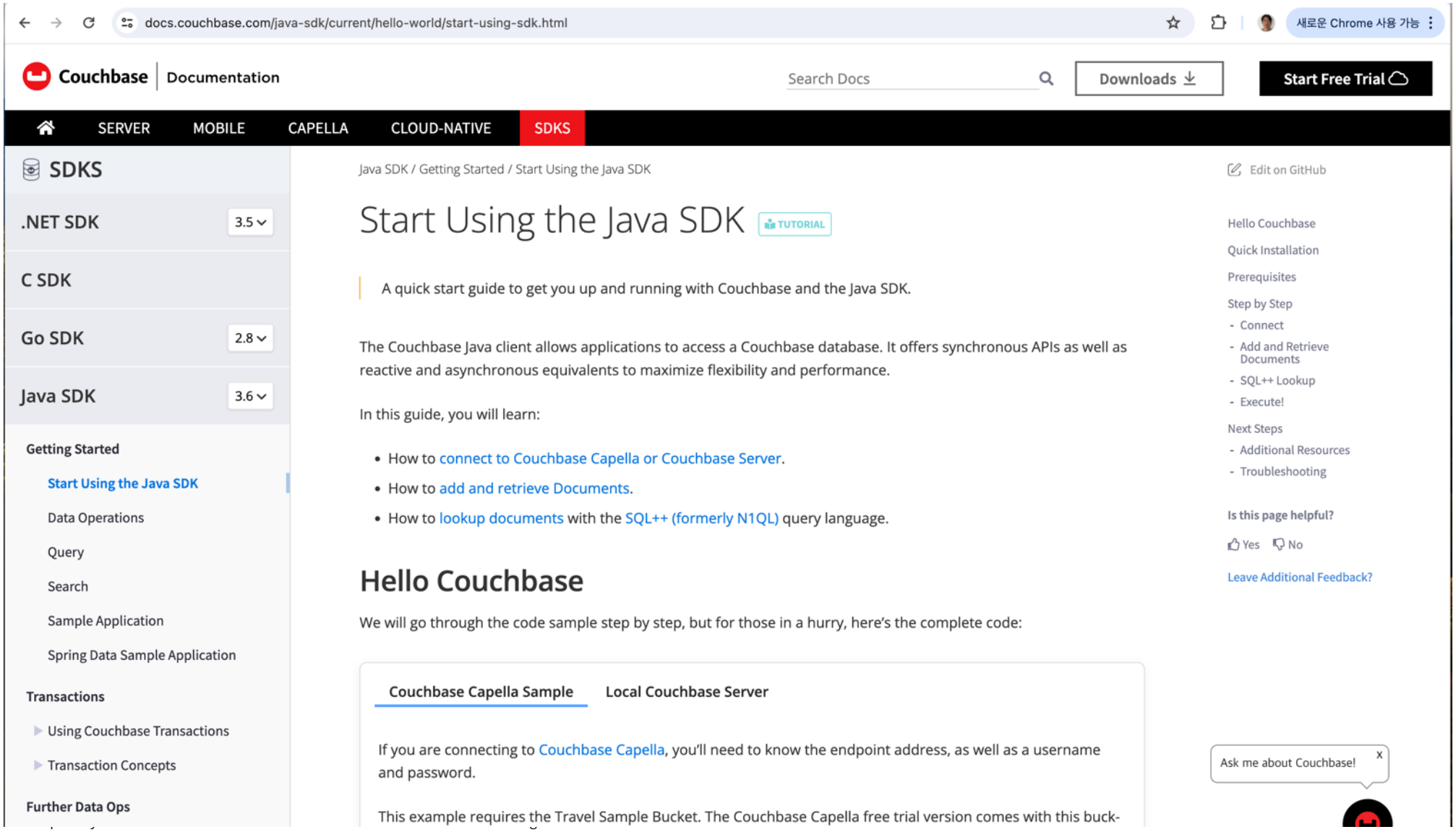
Couchbase Developer Tutorials

https://developer.couchbase.com/tutorials





Couchbase SDK Page



Semantic Search / GenAI Demo

Chat with PDF using LangChain
https://cb-chat-with-pdf.streamlit.app/
Password: Km2oTvY22vPC87

Chat with PDF using LlamaIndex
https://cb-rag-demo-llama-index.streamlit.app/
Password: Km2oTvY22vPC87

Chat with Couchbase Documentation
https://chat-with-cb-docs.streamlit.app/
Password: bcKPagE#FfJk$2

https://docs.couchbase.com/home/index.html

https://github.com/couchbase-examples/hybrid-search-demo

https://github.com/couchbase-examples/rag-demo

https://github.com/couchbase-examples/graphrag

Semantic Search Demo : 뉴스 추천

https://github.com/unixfree/semanticsearch4news

```
# 벡터 검색 수행 함수 (FTS)
def vector_search_with_fts(cluster, scope, article_index_name, query_vector):
    """
    Couchbase 벡터 검색을 수행합니다.
    :param cluster: Couchbase 클러스터
    :param scope: Couchbase 스코프
    :param article_index_name: FTS 인덱스 이름
    :param query_vector: 검색할 벡터
    """
    try:
        # 벡터 검색 쿼리 설정
        vector_search = VectorSearch.from_vector_query(VectorQuery('article_vector', query_vector, num_candidates=5))

        request = search.SearchRequest.create(vector_search)

        # 검색 수행
        result = scope.search(article_index_name, request)

        print(f"FTS Vector Search results:")
        for row in result.rows():
            print(f"ID: {row.id}, Score: {row.score}")
            doc = collection.get(row.id)
            doc_content = doc.content_as[dict]  # 문서를 사전 형식으로 변환
            print(f"Title: {doc_content['title']}")
            print(f"Date: {doc_content['date']}")
            print(f"Url: {doc_content['url']}")
            print("--------")

    except CouchbaseException as e:
        print(f"Search failed: {e}")
```

Semantic Search Demo : 뉴스 추천

```
# SQL++ 하이브리드 검색 수행 함수
def hybrid_vector_search_with_sql(cluster, article_vector, title_vector, title_text):
    """
    Couchbase SQL++, 자연어검색, 벡터 검색을 결합하여 검색을 수행합니다.
    :param cluster: Couchbase 클러스터
    :param article_vector: 검색할기사 내용 벡터
    :param title_vector: 검색할기사 제목 벡터
    :param title_text: 검색할 단어
    """
    try:
        # N1QL을 사용한 KNN 및 필터 검색
        query = f"""
        SELECT title, date, author, url, like_count, SEARCH_SCORE() AS score
        FROM `news`.naver.article AS t1
        WHERE author like "%기자"
        AND like_count >= 1
        AND SEARCH(t1, {{
              "query": {{"match": "{title_text}","field":"title"}}
            }})
        AND SEARCH(t1, {{
              "query": {{"match_none": {{}}}},
              "knn": [{{"field": "article_vector", "vector": {article_vector}, "k": 5}}],
              "knn": [{{"field": "title_vector", "vector": {title_vector}, "k": 5}}]
            }})
        ORDER BY score,date DESC
        """

        # 쿼리 실행
        result = cluster.query(query)
```

```
# 결과 출력
print("")
print(f"SQL++ Hybrid Search results:", result)
for row in result:
    print(f"Score: {row['score']}")
    print(f"Title: {row['title']}")
    print(f"Date: {row['date']}")
    print(f"Author: {row['author']}")
    print(f"Like Count: {row['like_count']}")
    print(f"Url: {row['url']}")
    print("-------")
except CouchbaseException as e:
    print(f"Hybrid search failed: {e}")
```

Retrieval-Augmented Generation (RAG)

https://github.com/jon-strabala/easy-webrag-langchain-demo

- Create and activate a virtual environment in a new empty demo directory

```
$ mkdir MYDEMO
$ cd MYDEMO
$ python3 -m venv .venv
$ source .venv/bin/activate

$ git clone https://github.com/jon-strabala/easy-webrag-langchain-demo.git
$ cd easy-webrag-langchain-demo
```

- Install dependencies

```
$ pip install -r requirements.txt
```

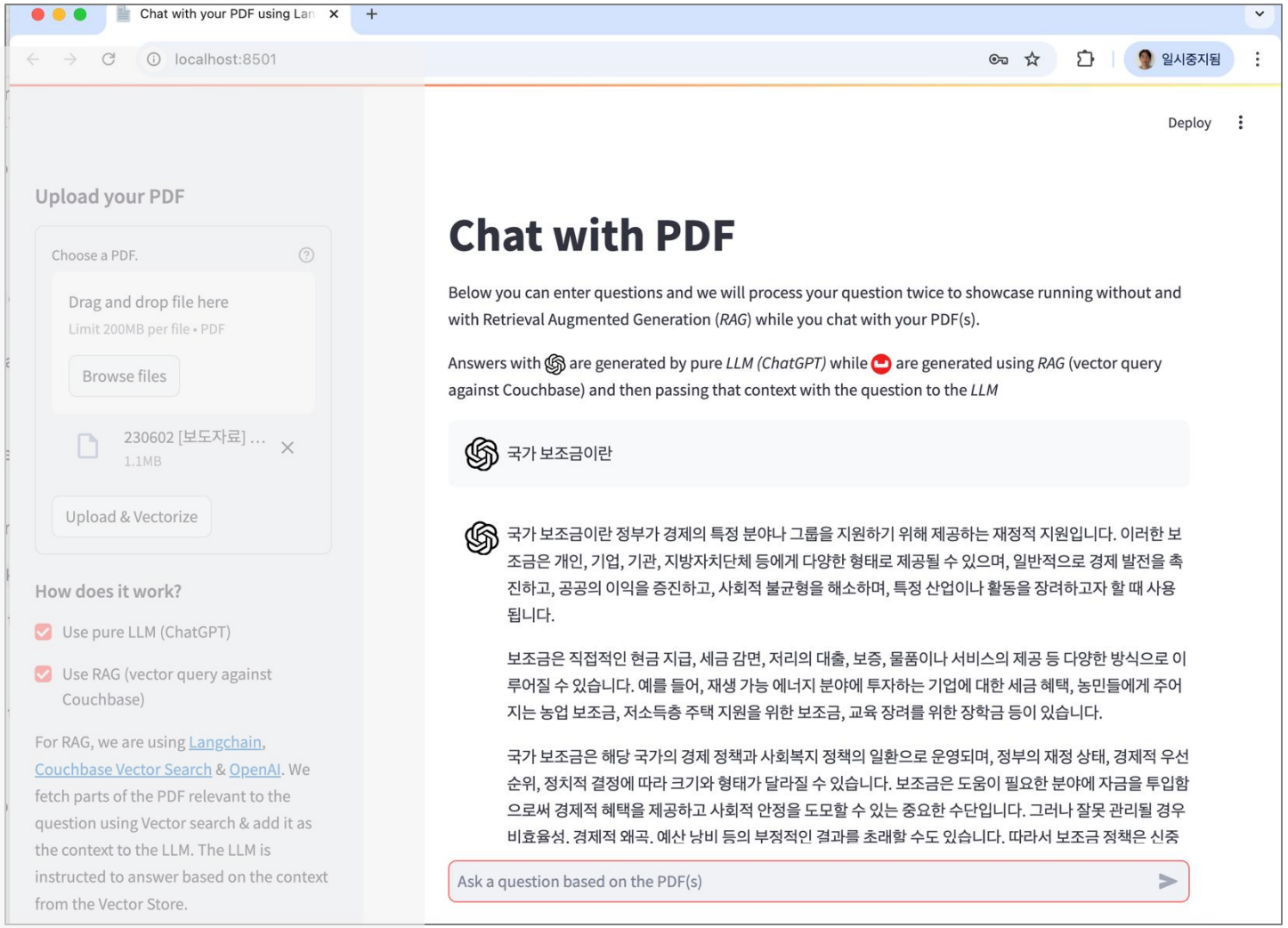
- Copy the template environment template

```
$ cp _setup.tmpl _setup

$ vi _setup
  export CB_BUCKET=vectordemos
  export CB_SCOPE=langchain
  export CB_COLLECTION=webrag
  export CB_SEARCHINDEX=webrag_index

$ source _setup
$ ./check_couchbase.sh
$ ./check_openai.py
$ ./setup.py

### Run the application this will start a webserver
$ streamlit run chat_with_pdf.py
```



And then …

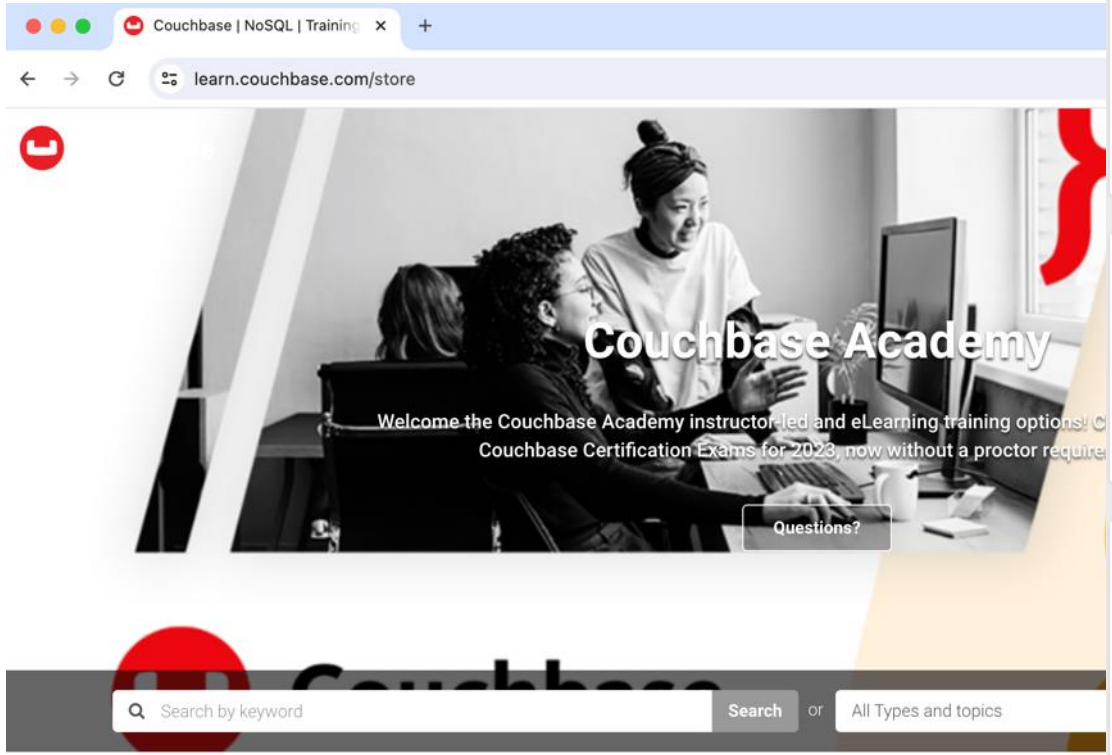
Next Capella Webinar

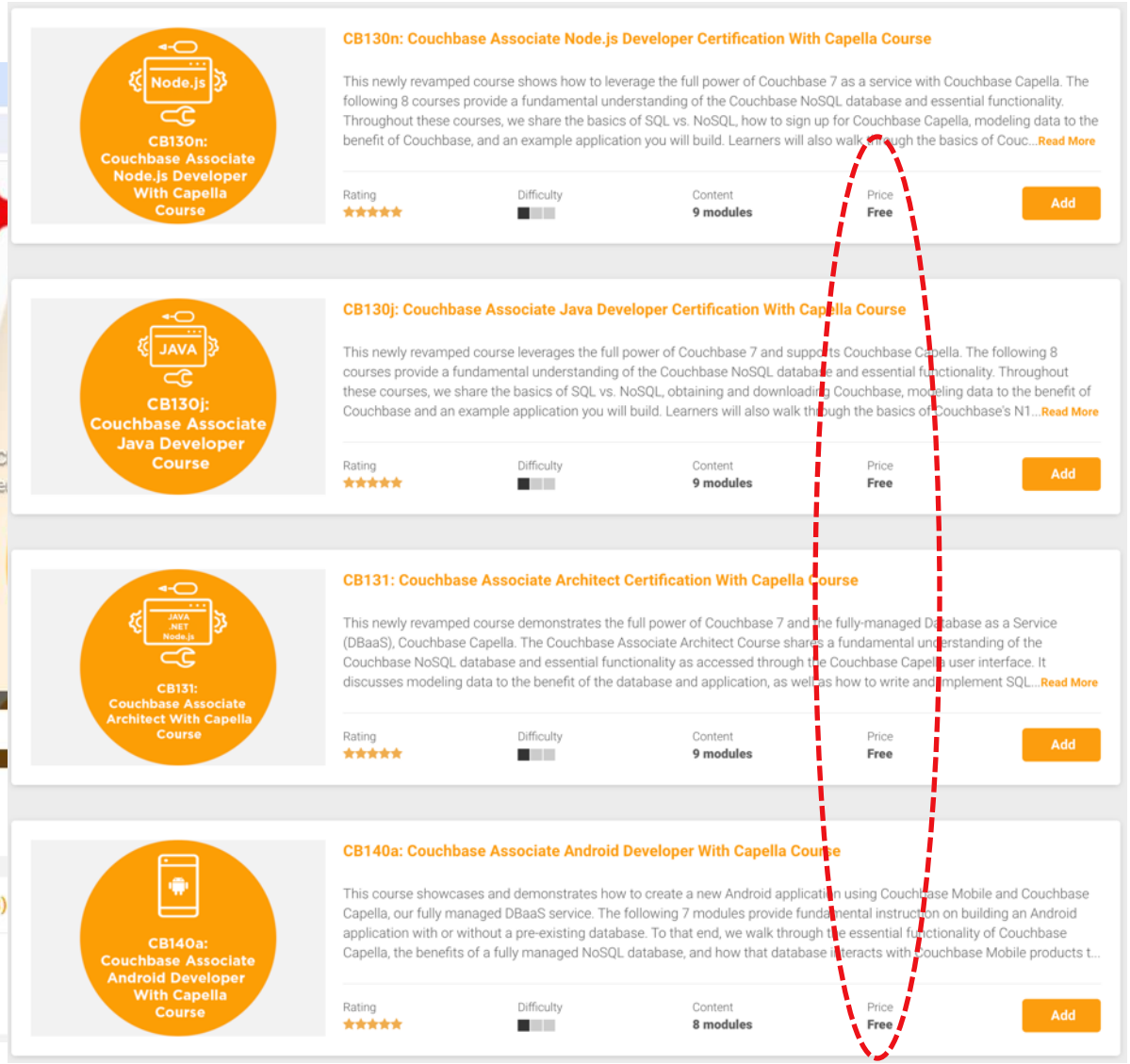
구분	일자	Webinar 주제
시리즈 1	2024-09-25	벡터 검색을 활용한 AI Powered 어플리케이션 구축
시리즈 2	2023-10-30	벡터 검색을 활용한 GenAI(LLM/RAG) 어플리케이션 구축
시리즈 3	**2024-11-27**	**벡터 검색을 활용한 Mobile On-Device AI 어플리케이션 구축**

Couchbase Academy

https://docs.couchbase.com/
https://developer.couchbase.com/
https://couchbase.live/
https://query-tutorial.couchbase.com/tutorial/#1

https://learn.couchbase.com/store





Thank you!

Paul.Son@couchbase.com

www.couchbase.com

cloud.couchbase.com

Couchbase