OTUS

Онлайн образование

otus.ru

Проверить, идет ли запись

Меня хорошо видно && слышно?

Couchbase

Аристов Евгений

telegram @AEugene

https://aristov.tech

Правила вебинара

Активно участвуем

Задаем вопрос в чат

Вопросы вижу в чате, могу ответить не сразу

Маршрут вебинара

Цели вебинара

1 Понять зачем используется Couchbase

2 Познакомиться с архитектурой Couchbase

Смысл | Зачем вам это уметь, в результате:

1 Выбрать оптимальный вариант NoSQL решения

2 Уметь базово настроить Couchbase

Couchbase

The Modern Database for Enterprise Applications


Orbitz
TripAdvisor
Rentalcars
United
Marriott
Concur
Amex
Visa
PayPal
Experian
The foundation of all
these apps is
a Couchbase database
Ebay
Staples
Louis Vuitton
Tesco
Carrefour
Disney
Hulu
Nielsen
Verizon

A Proven Enterprise Solution Chosen by Industry Leaders

Retail & E-Commerce

Walmart
cars.com
Fanatics
Office DEPOT OfficeMax
STAPLES
TESCO
QVC
BEST BUY

3 of the Top 103
eCommerce Companies

Travel & Hospitality

Marriott
amadeus
Sabre
ORBITZ
tripadvisor
skyscanner

of the Top 3
GDS Companies

Financial Services

ADP
FICO
CONCUR
PayPal
WELLS FARGO
VISA

3 of the Top 3
Credit Reporting Companies

Healthcare

WebMD
everyday HEALTH
BD
Cochlear
MONSENSO

3 Fortune 500
Healthcare Companies

Telecom

COMCAST
KDDI
at&t
Viber
Telefónica

Media & Entertainment

sky
THOMSON REUTERS
FANDANGO
GANNETT
Dushkin McGraw-Hill
Disney
nielsen

6 of the Top 10
Broadcast Companies

Gaming

EA
betfair
zynga
BLIZZARD
Playtika
SGN

6 of the Top 10
Online Casino Gaming Companies

Industrial IoT

verizon
Numerex
CHAMBERLAIN
GE

2 of the Top 2
IoT Platforms

Couchbase Behind Today's Business-Critical Applications

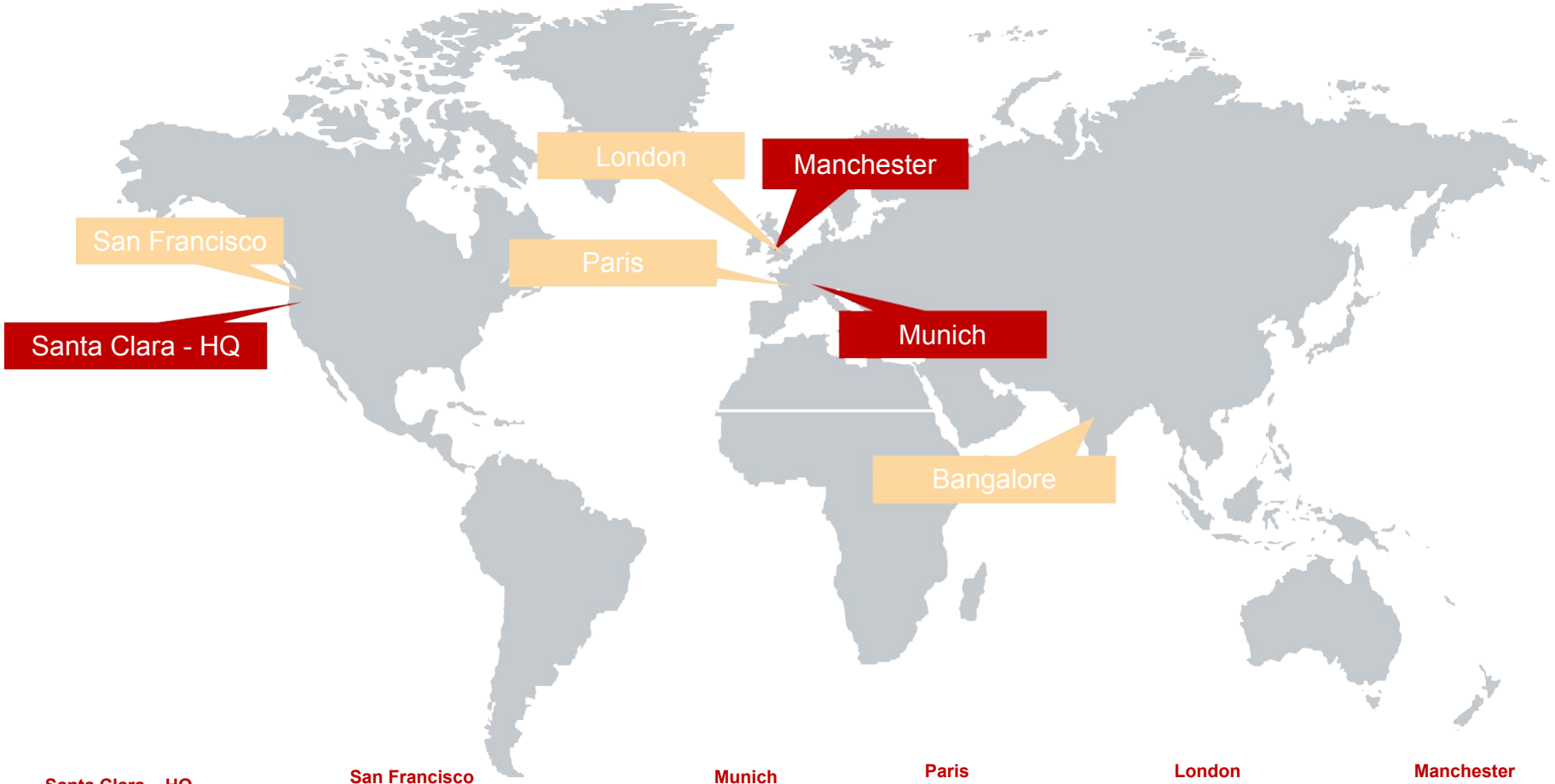
	LinkedIn	TESCO	amadeus	COMCAST	UNITED
Customers					
Application	Caching & session store for single view	Real-time pricing, product catalog, inventory management	Flight availability, booking, pricing analytics, etc.	Customer 360 single view, unified notes	Real-time crew management, scheduling and resources
Performance	**2M+** reads/sec. **10M** queries/sec.	**10M+** unique SKUs **35K** requests/sec.	**15M** ops / second **<2.5ms** response time	**210M** documents **100K** users	**41K** pilots and crew **148M** travelers in 2017

Infrastructure

Developer Agility | **Performance at Scale** | **Manageability**

Security + Availability

Couchbase: Brief Company Overview



- Founded 2011
- ~ 700 employees
- 70%+ new business growth
- 100%+ growth in total contract value
- 55%+ billings growth
- 50%+ growth in average deal size
- IPO 22.07.2021

Santa Clara – HQ
3250 Olcott Street
Santa Clara, CA 95054
United States

San Francisco
450 Mission Street #200
San Francisco, CA 94105
United States

Munich
Prielmayerstr. 3
80335 Munich
Germany

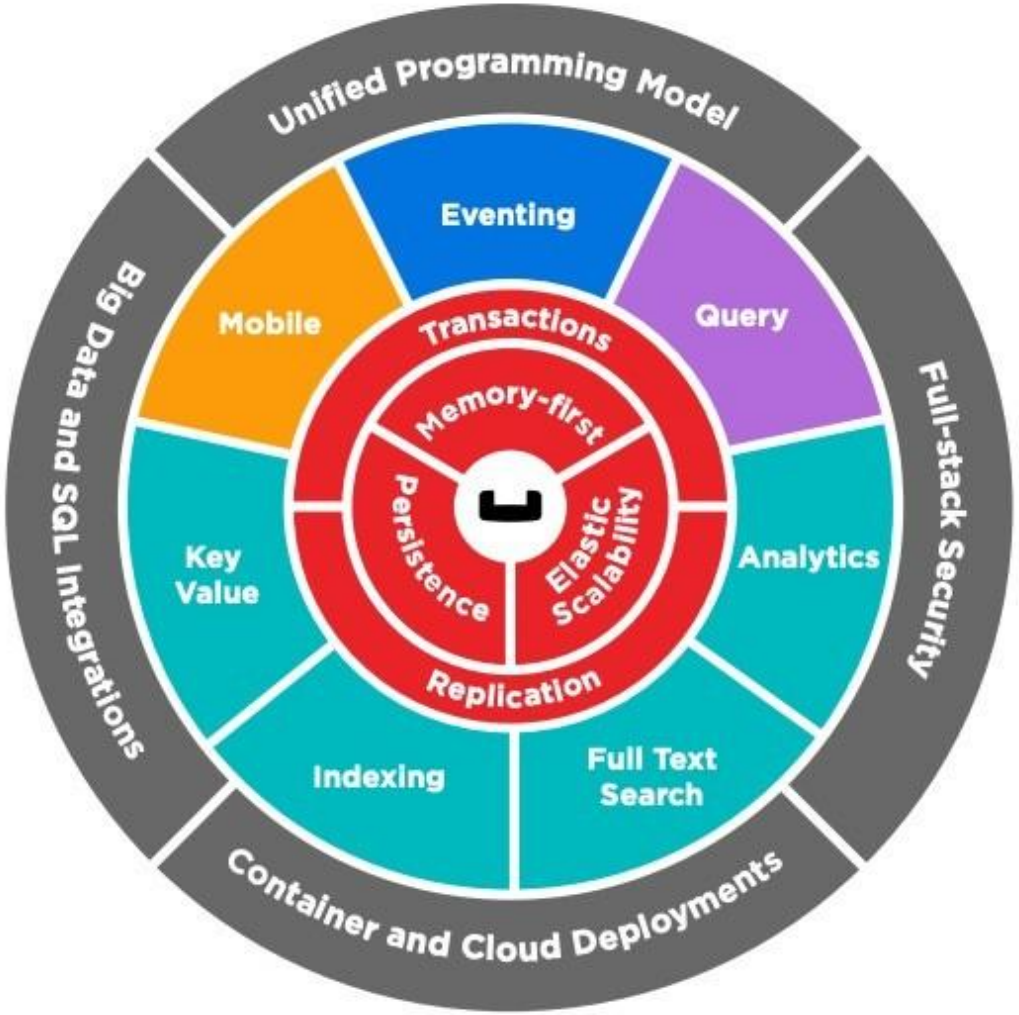
Paris
40 Rue du Collisée
75008 Paris
France

London
11-21 Paul Street
London EC2A 4JU
United Kingdom

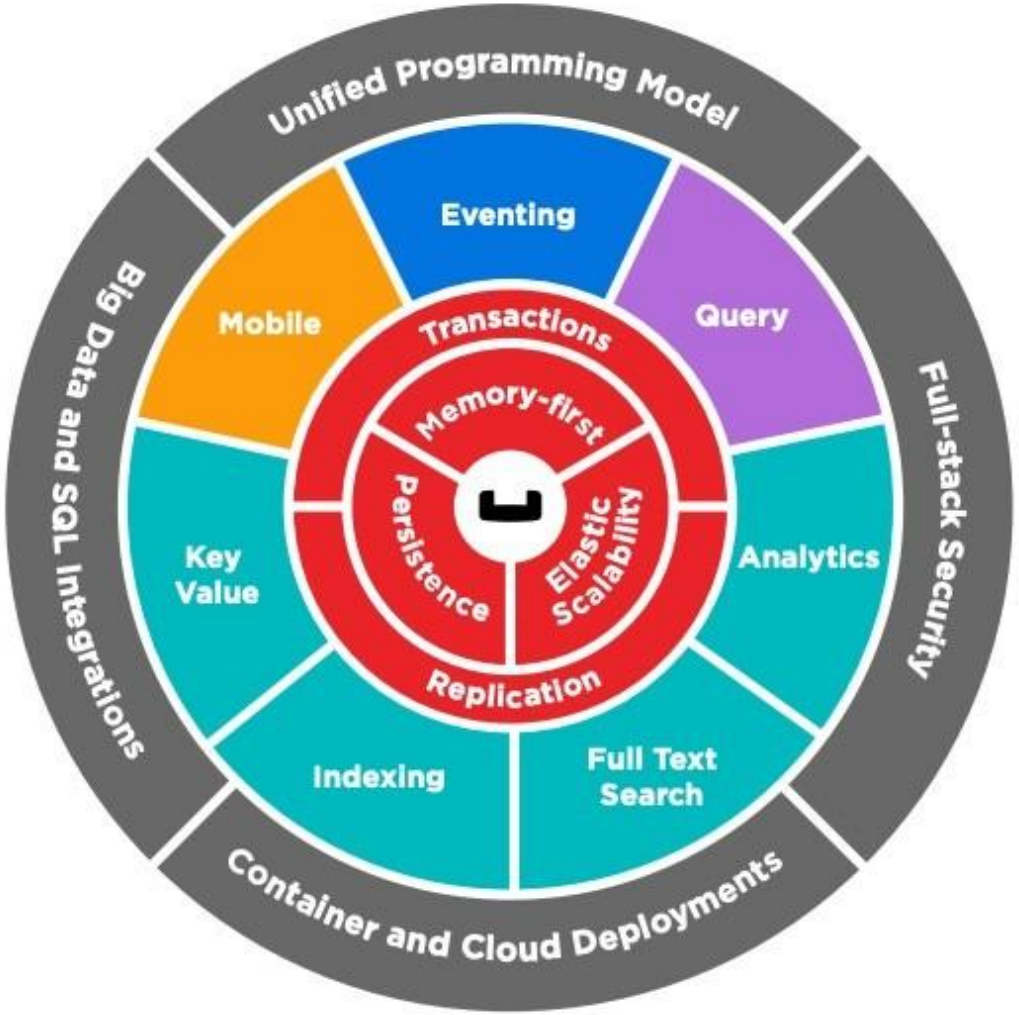
Manchester
1A Tariff St
Manchester M1 2FF
United Kingdom

Bangalore
10 Museum Road
Bangalore 560001
India

Architecture Overview

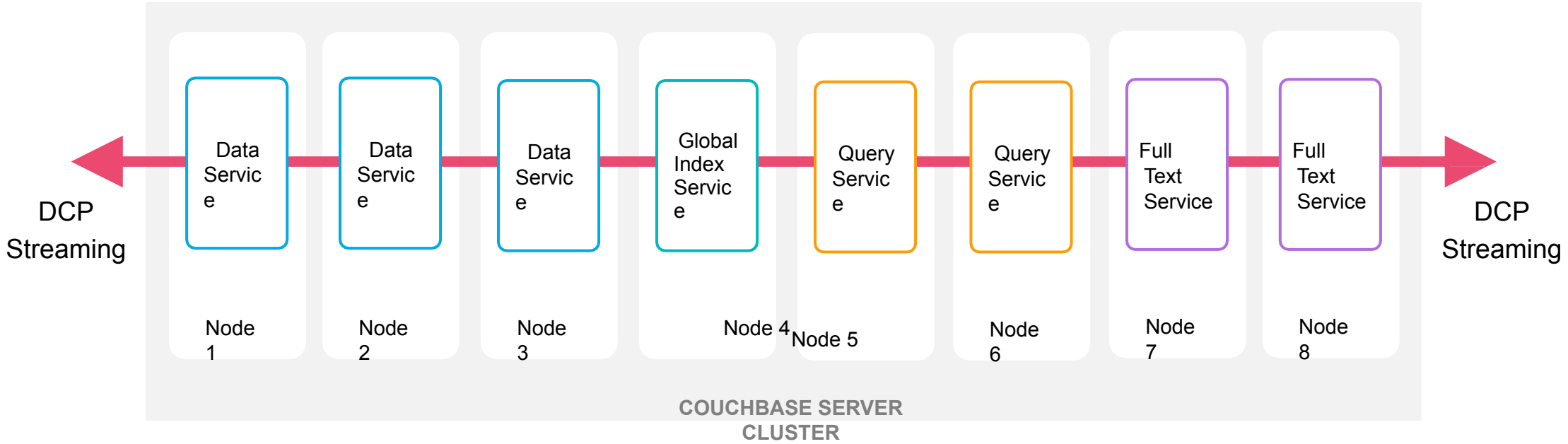


Memory-First



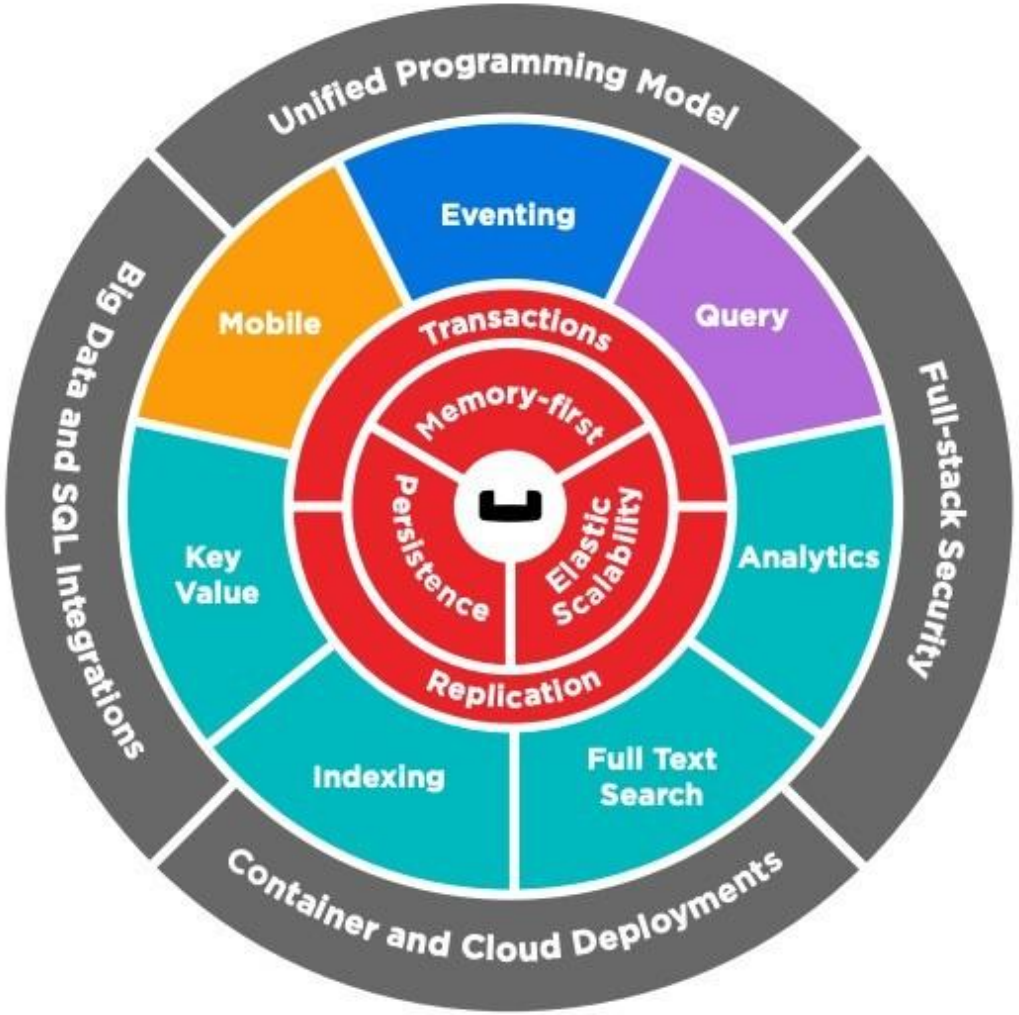
Memory-First Architecture

Data movement free from disk bottlenecks

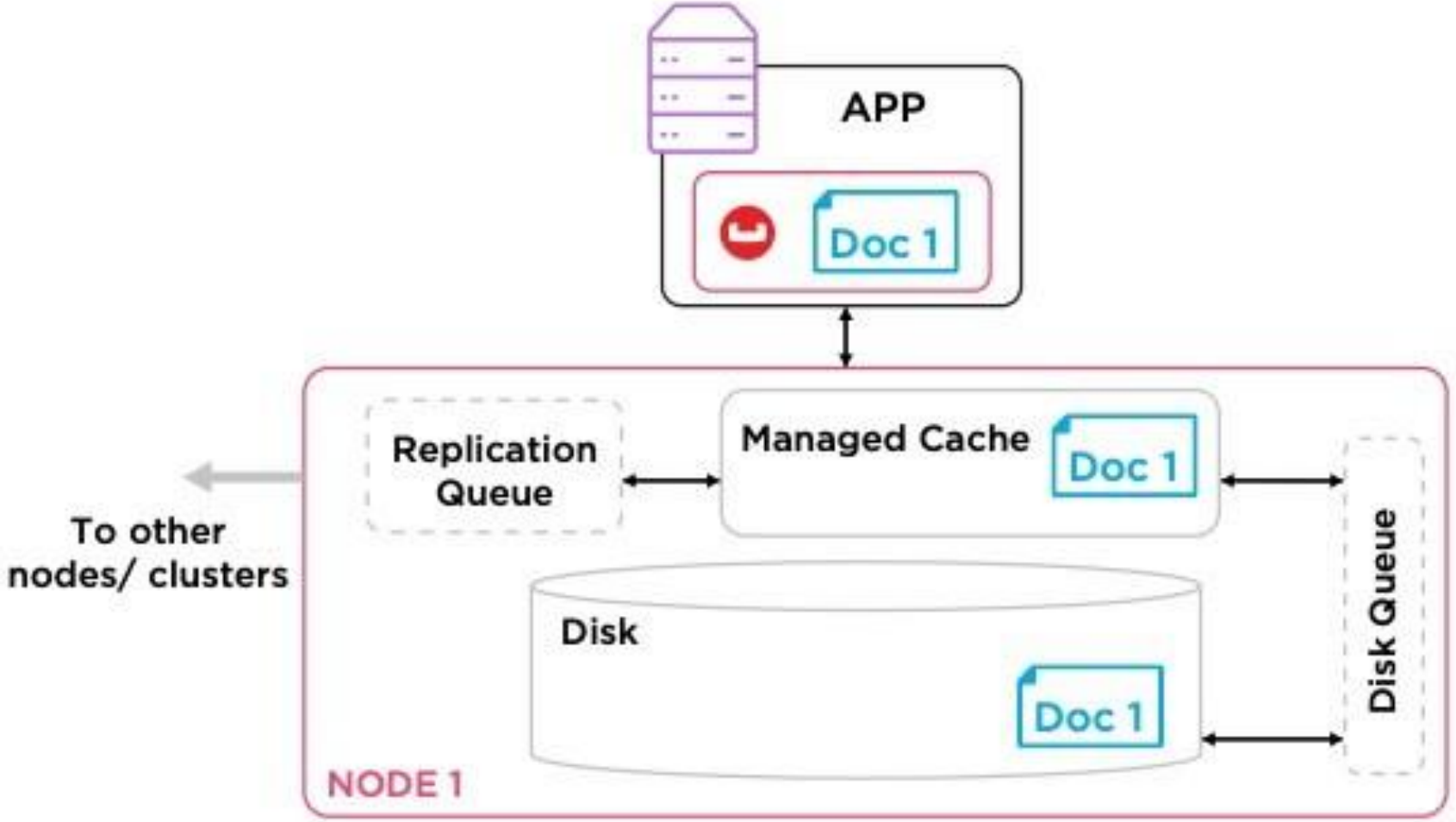


- In-memory streaming of updates to all components
- In-memory cache
- Memory-only data buckets
- Memory-only indexes

Persistence

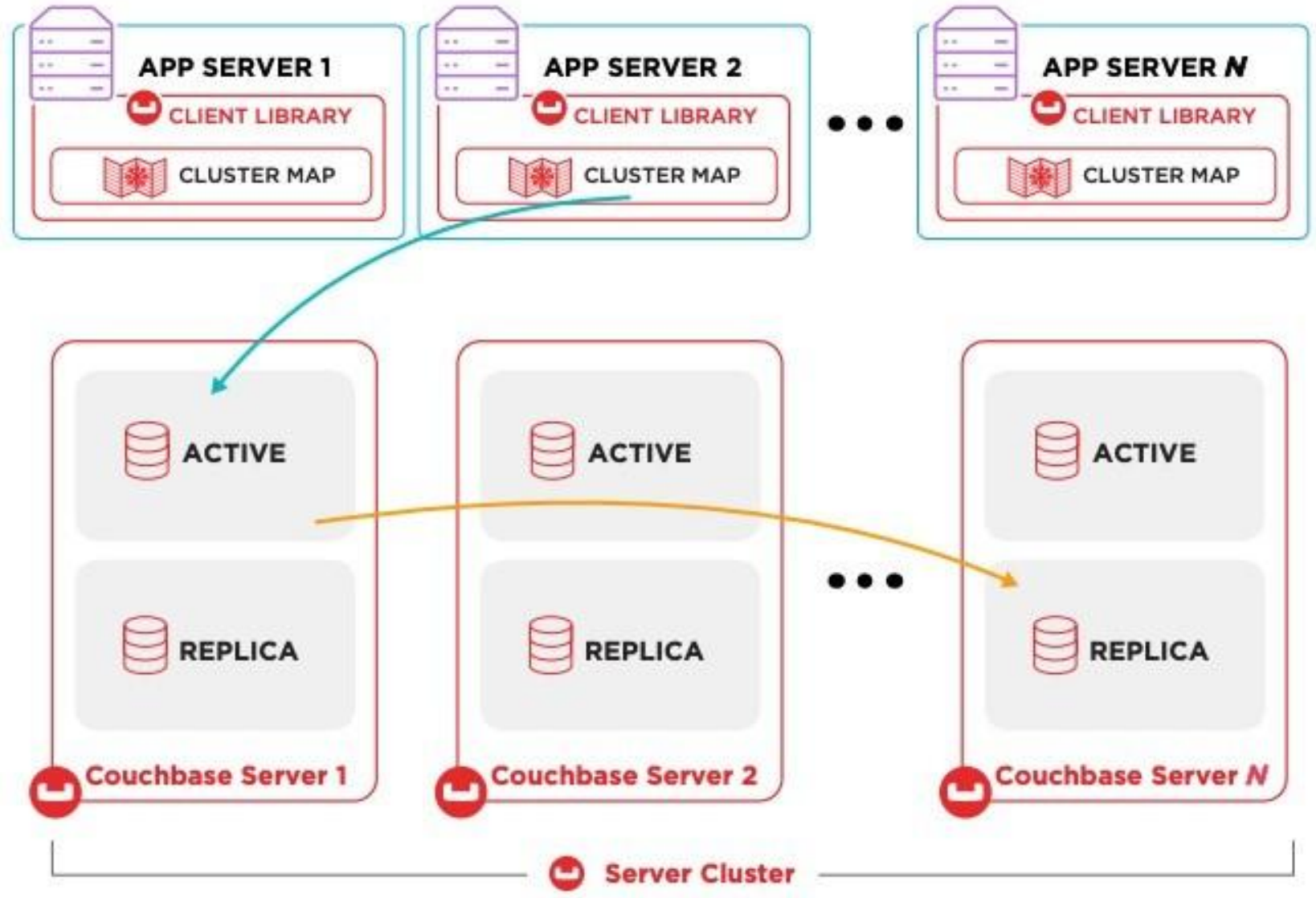


Persistence

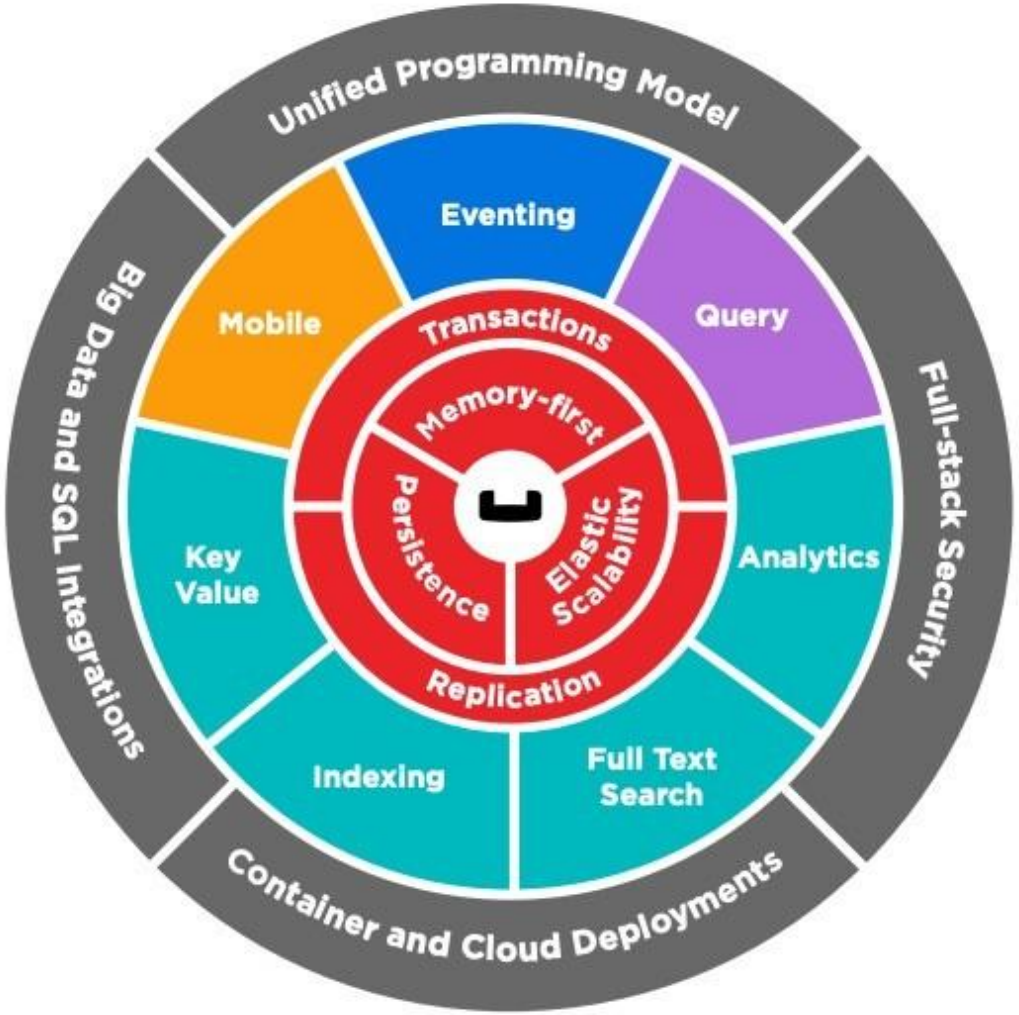


Couchbase Server Asynchronous Architecture

Persistence



Elastic Scalability



Multi-Dimensional Scaling

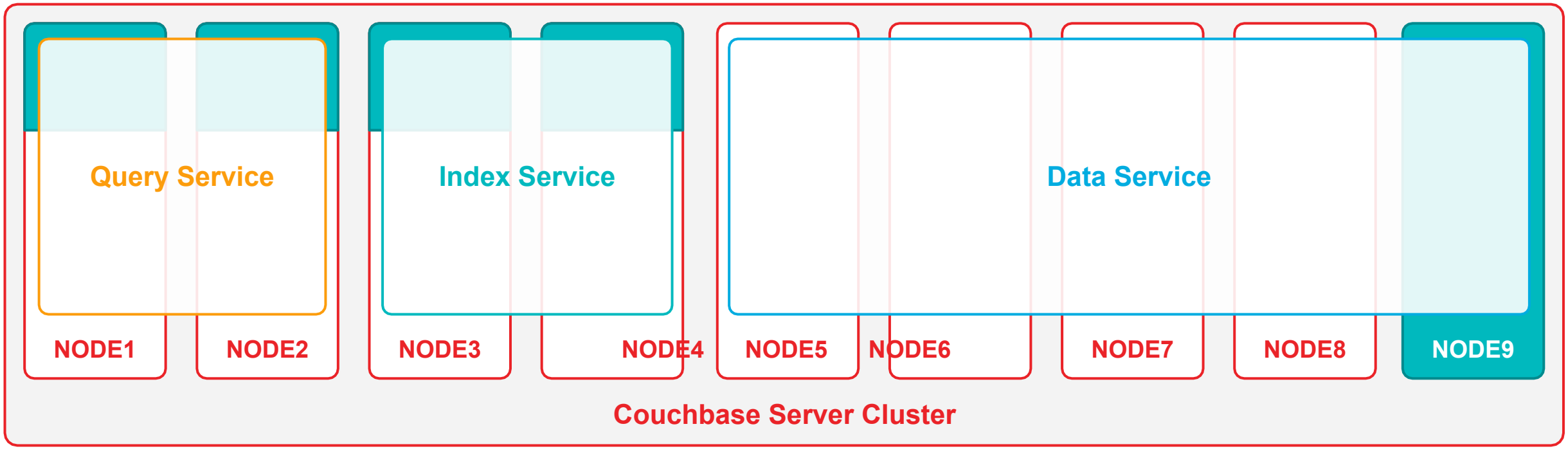
Independent Scalability for Best Computational Capacity – *per Service*

More CPU for Query Processing?

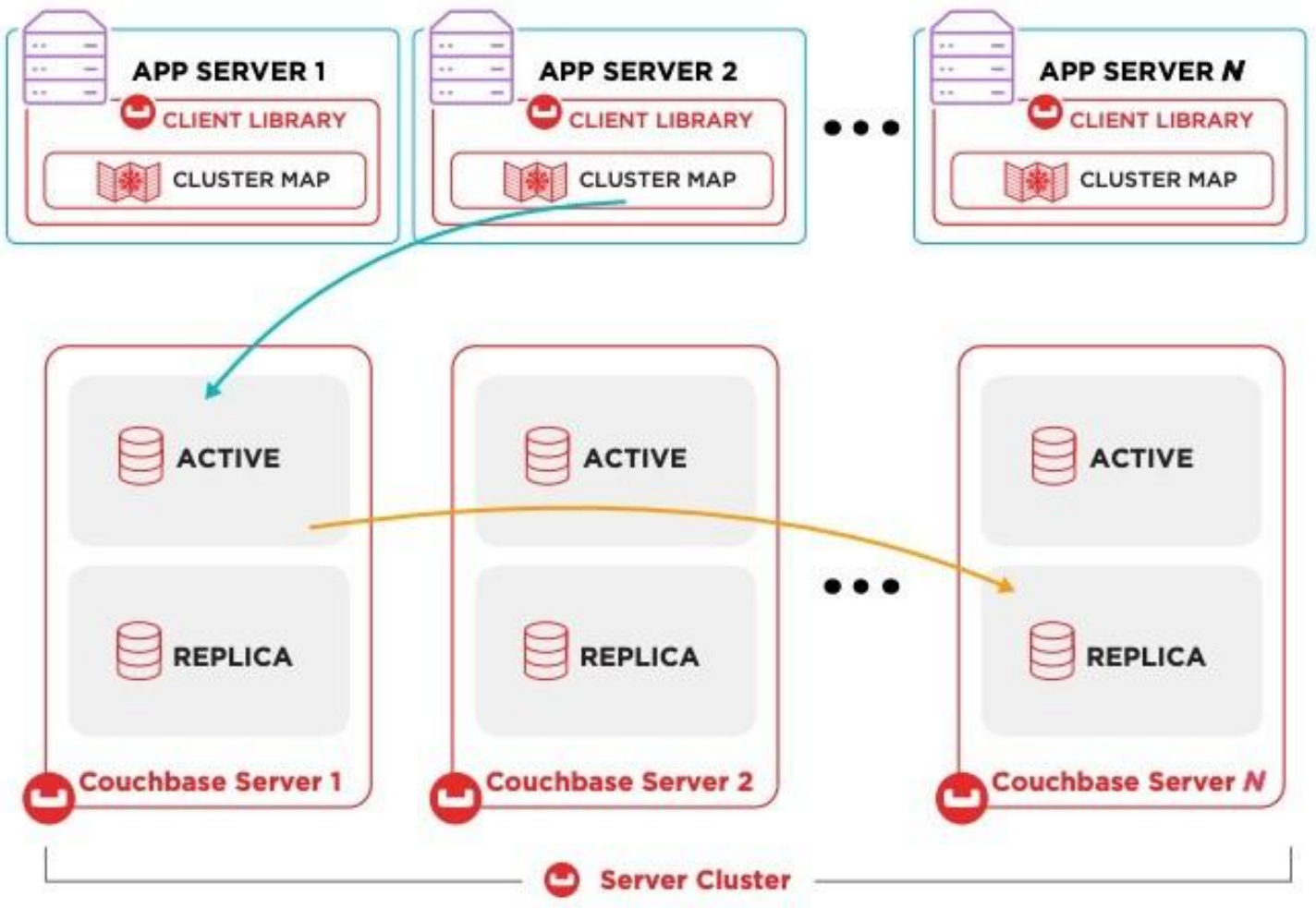
Scale up or out
Query Service Nodes.

Heavier Indexing?

Scale up or out
Index Service Nodes.

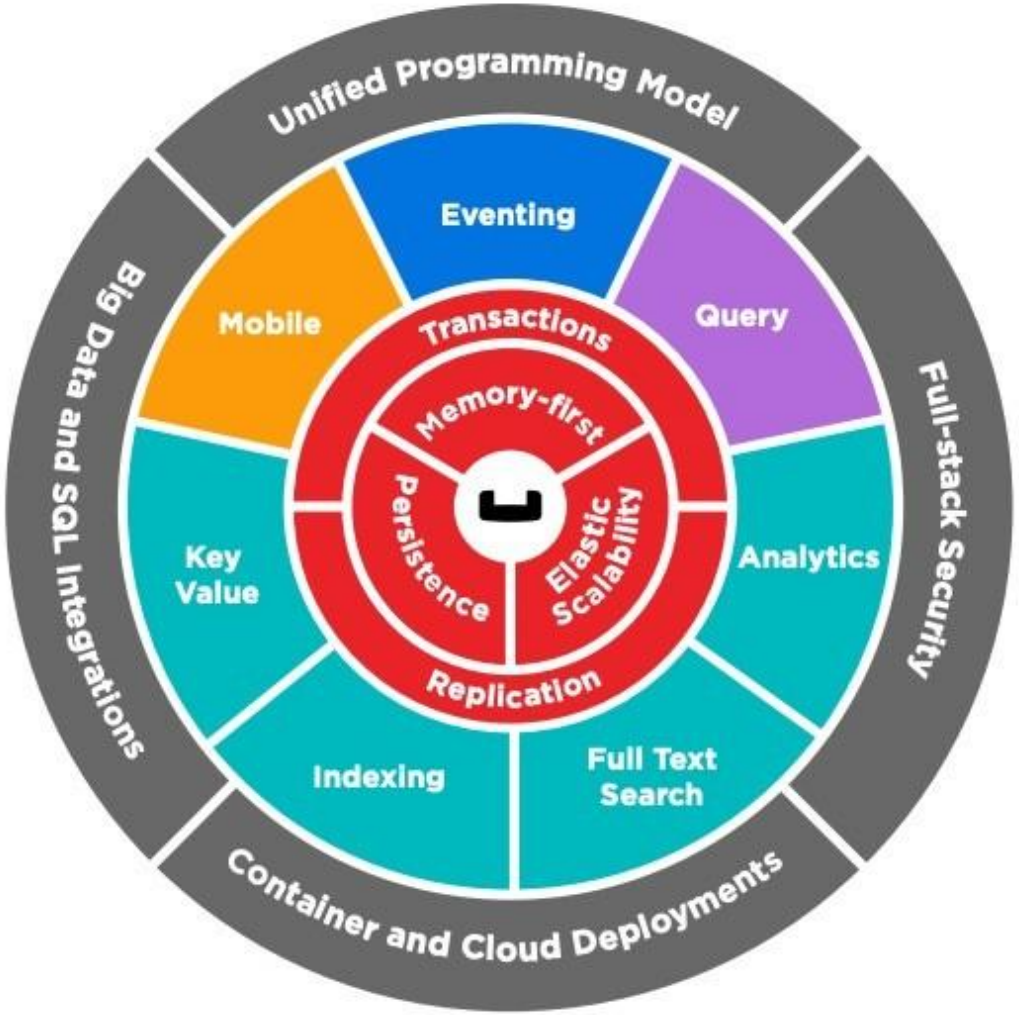


Elastic Scalability & High Availability

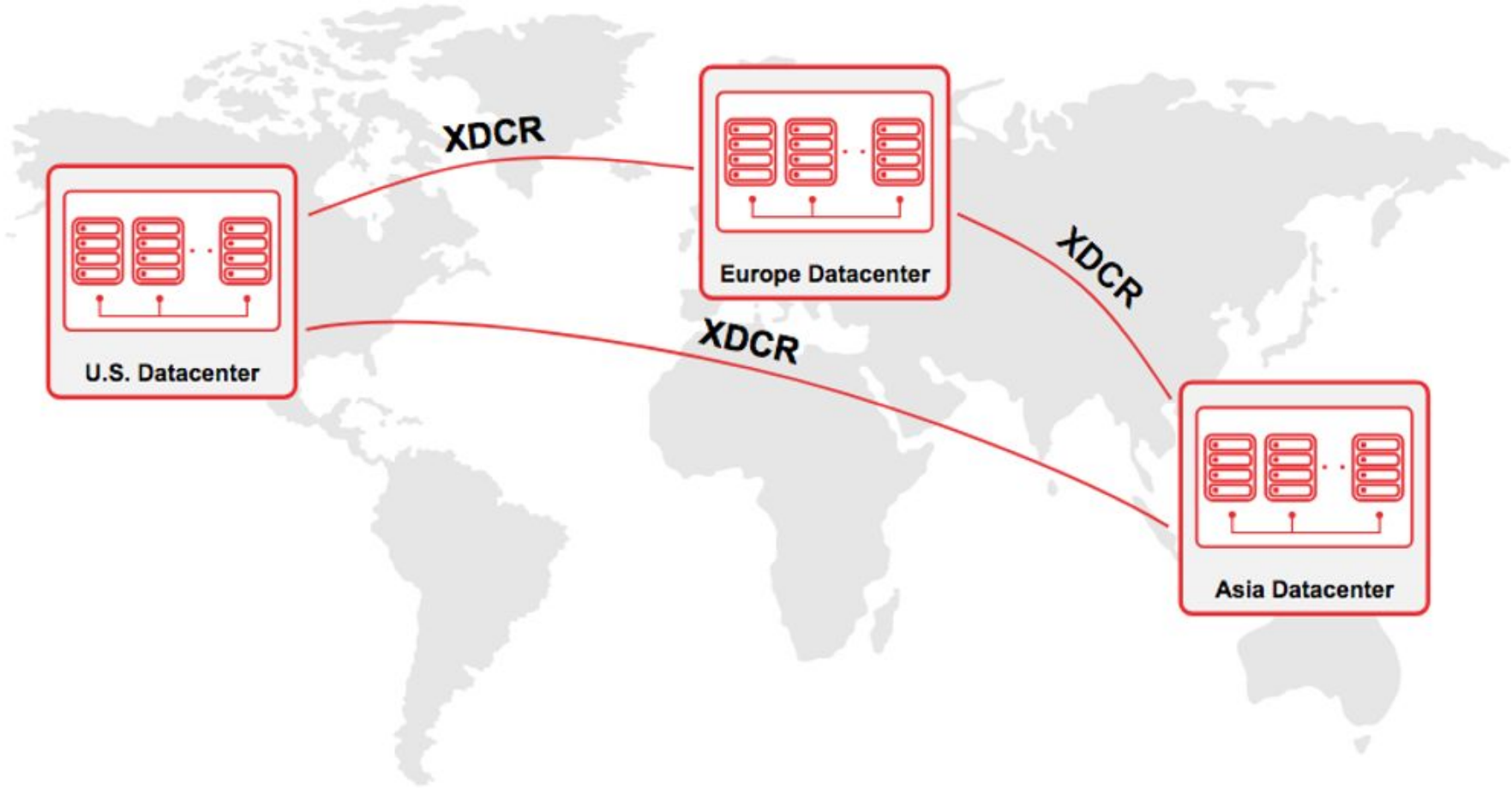


- **Scale on demand** with automatic partitioning and rebalancing
- **Build always-available apps** with memory-to-memory replication and automatic failover
- **Simplify development** with topology-aware clients and direct communication

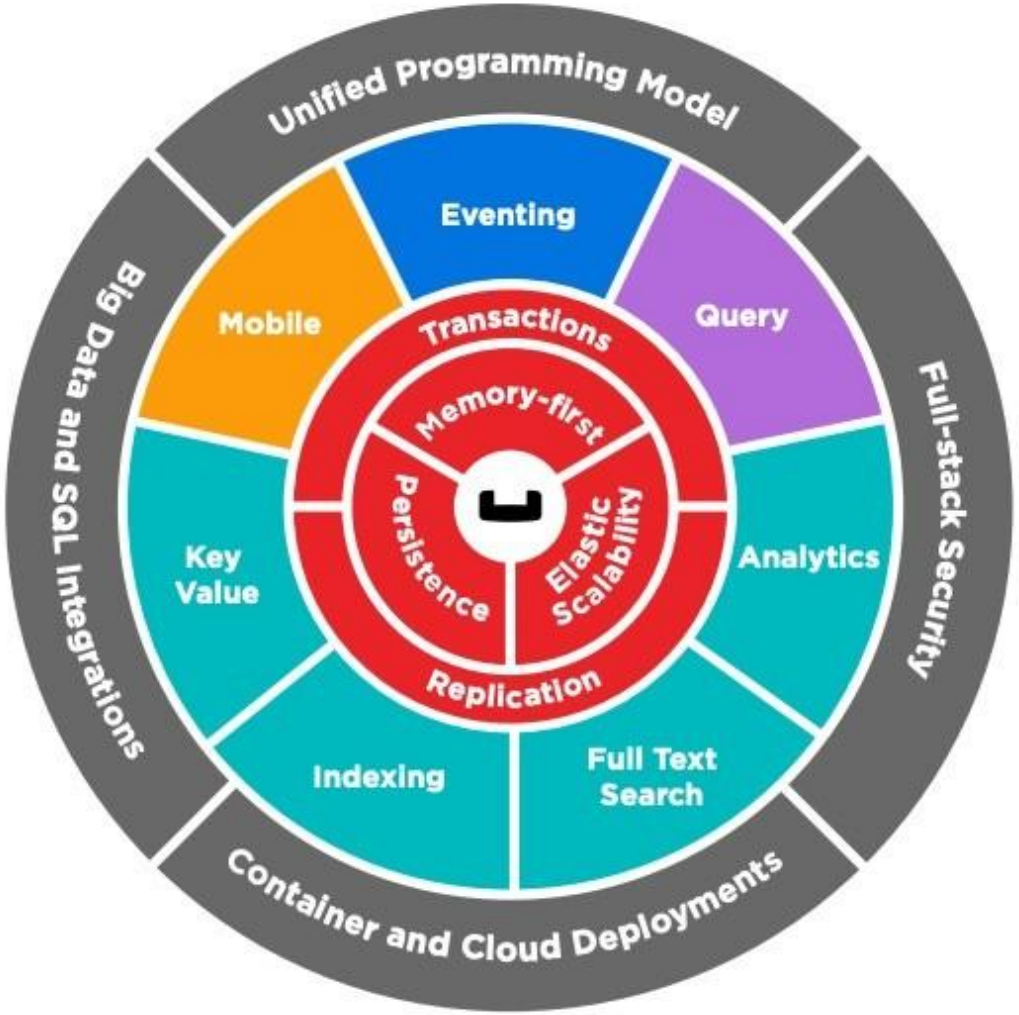
Replication



Cross Data Center Replication (XDCR)



Transactions



Scopes and Collections = Schemas and Tables

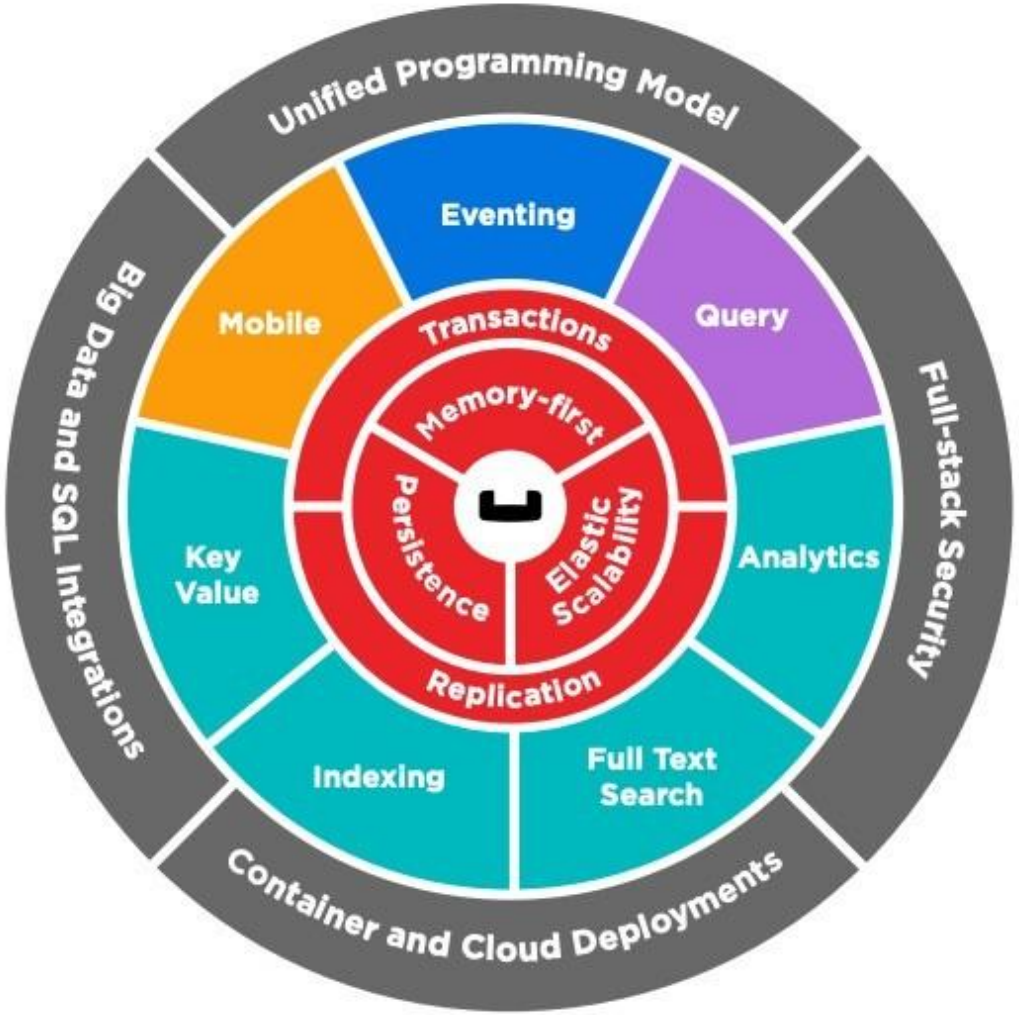
Relational Model	Couchbase
Server	Cluster
Database	Bucket
Schema	**Scope**
Table	**Collection**
Row	Document (JSON or BLOB)

Multi-Document ACID Transactions

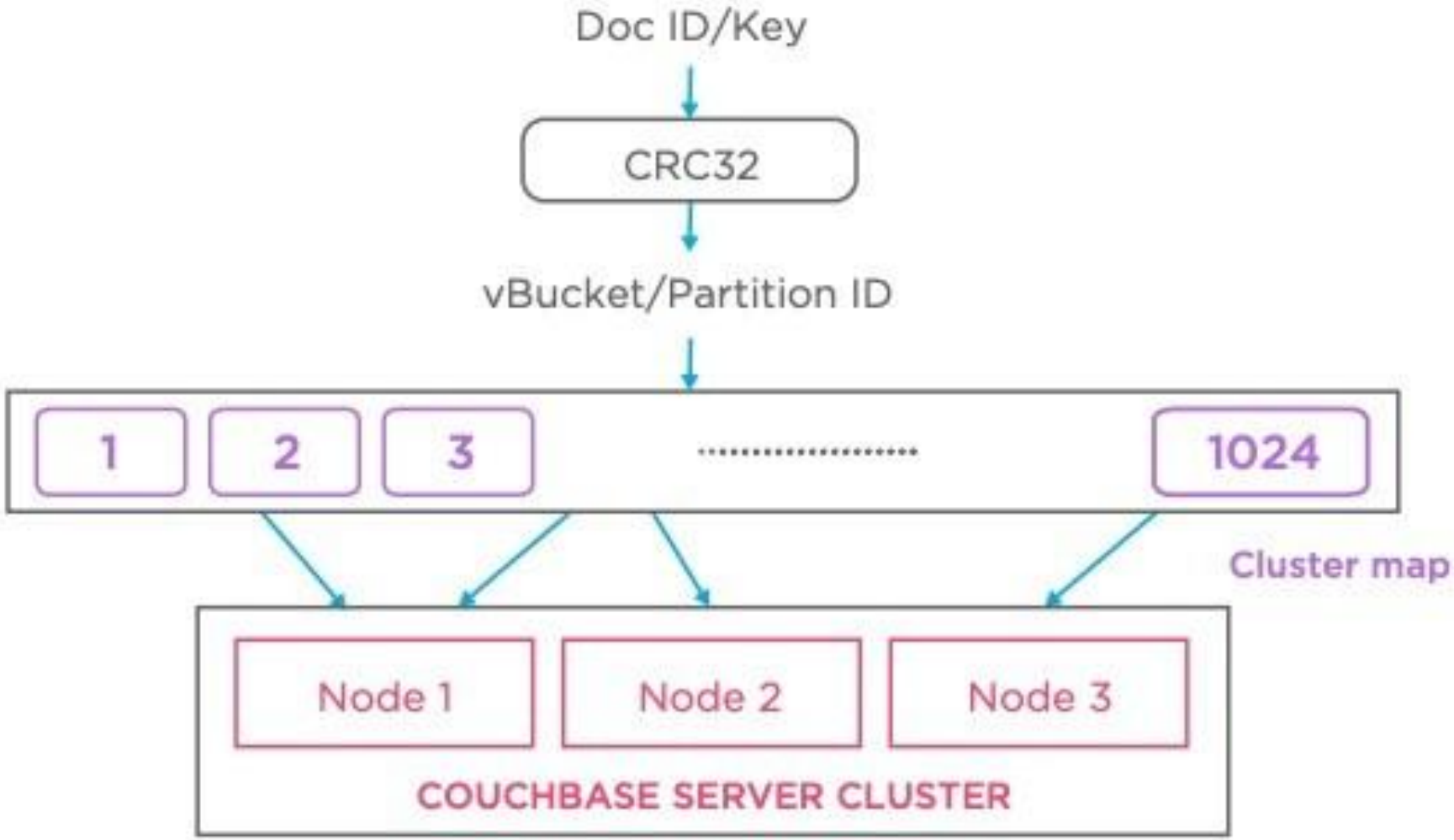
A	Atomicity	Guarantees all-or-nothing semantics for **updating multiple documents** in more than one shards on different nodes.
C	Consistency	**Replicas** immediately consistent with the transaction commit. **Indexes and XDCR** eventually consistent with the transaction commit (*N1QL can enforce strong consistency upon read with request_plus*)
I	Isolation	**Read Committed** isolation for concurrent transactions
D	Durability	**Data protection under failures**: 3 different levels - *replicate to majority of the nodes; replicate to majority and persist to disk on primary; or persist to disk on majority of the nodes.*

```
START TRANSACTION;
UPDATE customer SET balance = balance + 100 WHERE cid = 4872;
SELECT cid, name, balance  from customer;
UPDATE customer SET balance = balance - 100 WHERE cid = 1924;
SELECT cid, name, balance FROM customer;
COMMIT ;
```

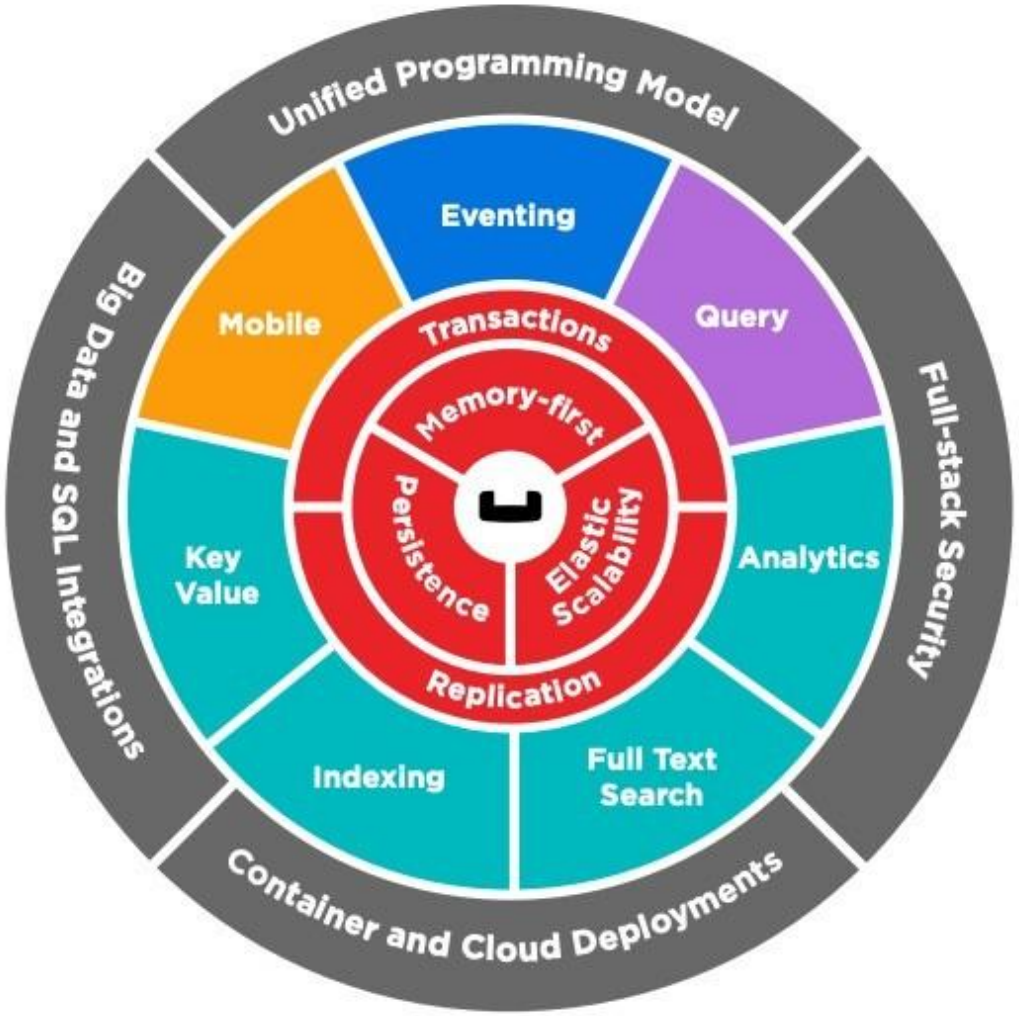
Key / Value


Unified Programming Model
Full-stack Security
Container and Cloud Deployments
Big Data and SQL Integrations
Eventing
Query
Analytics
Full Text Search
Indexing
Key Value
Mobile
Transactions
Memory-first
Persistence
Elastic Scalability
Replication

Key / Value Access



Indexing / Query



N1QL Origins

Named for ‘Not 1st Normal Form Query Language’

First released with Couchbase 4.0, Oct. 2015

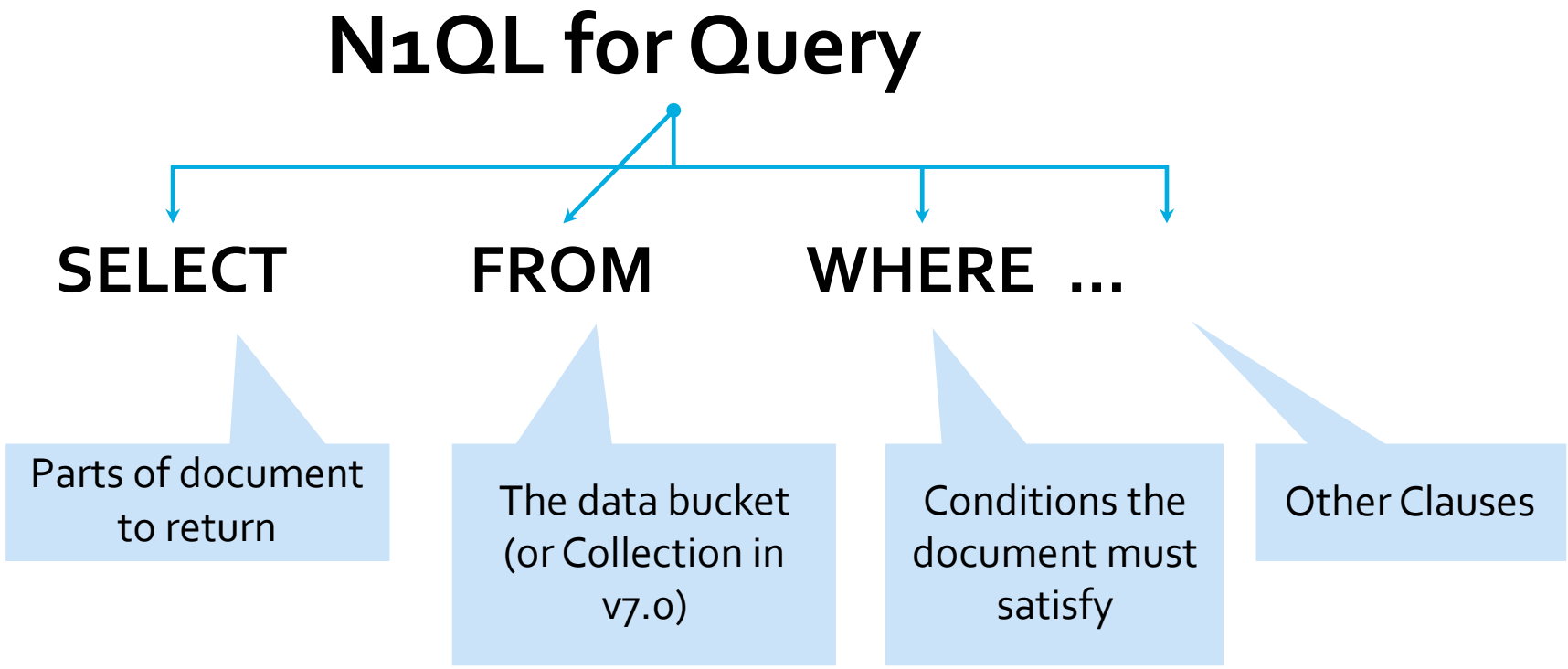
Work against flexible JSON Documents, not relational tables

Designed to be as close to SQL as possible

Overseen by Don Chamberlin, co-inventor of SQL

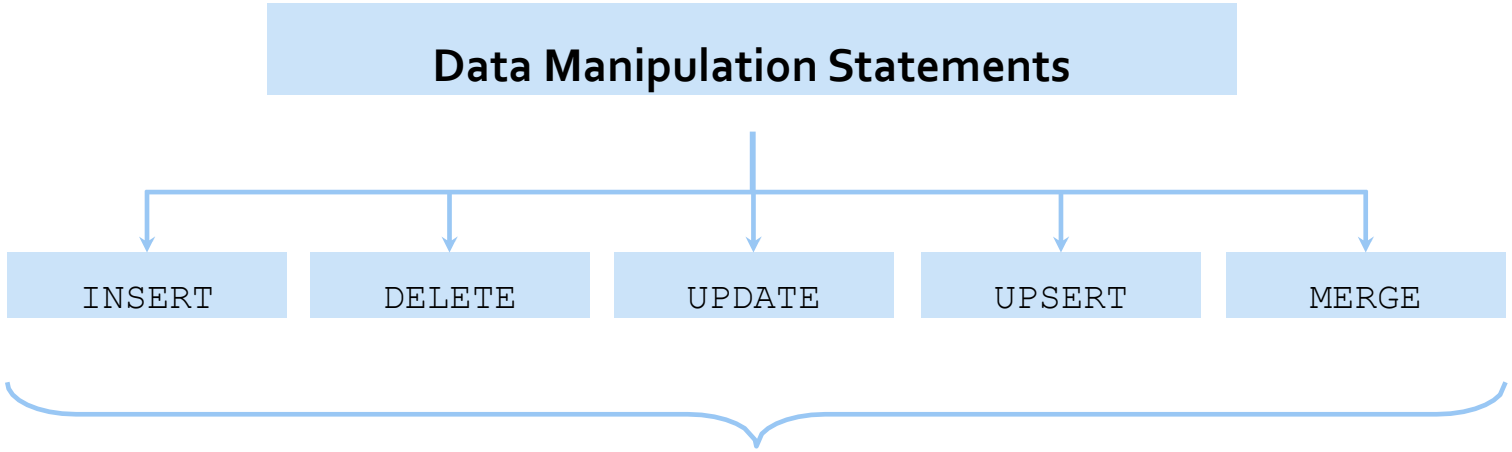
Power of SQL

Looks familiar ?



Power of SQL

Includes DML



These statements allow you to create, delete, and modify the **data stored in JSON documents** by specifying and executing simple commands

Power of SQL

Key Features

Indexes

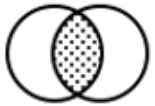
Use the CREATE INDEX and DROP INDEX statements to easily create, and delete indexes

Access Path

Traverse embedded documents with simple ‘dot’ notation

Aggregation

Provides aggregation operators, such as MIN, MAX, COUNT as well as grouping operators, the GROUP BY clause, and the group filter, HAVING



Joins

Retrieve data from multiple documents spanning one or more buckets

Nested Queries

Query within another query, embedded within the where clause

Power of SQL

Extends SQL to work with Array

Nesting

NEST lets you retrieve sub-records for a record

That is, for each left hand input, the matching right hand inputs are collected into an array, which is then embedded in the result

Unnesting

UNNEST is the opposite of NEST

extracts nested sub-records from a record, making each extraction a separate record. Unnesting is performed on a single document

Raw

SELECT RAW converts multiple results into an array (not a document)

Power of SQL

Support Joins

JOIN TYPE	Examples
LOOKUP JOIN	SELECT .. FROM customer c **JOIN** orders o **ON KEYS** ['..']
INDEX JOIN	SELECT .. FROM customer c **JOIN** orders o **ON KEYS** o.customer_id
ANSI JOIN	SELECT .. FROM customer c **JOIN** orders o **ON** o.customer_id = c.id
ANSI JOIN Complex	SELECT .. FROM `travel-sample` airline **JOIN** `travel-sample` route **ON** route.airlineid = "airline_" || tostring(airline.id) **AND** route.type = "route"
ANSI JOIN with IN CLAUSE	SELECT .. FROM `travel-sample` route **JOIN** `travel-sample` airport **ON** airport.faa **IN** [route.sourceairport, route.destinationairport] **AND** airport.type = "airport"
ANSI LEFT OUTER	SELECT .. FROM `travel-sample` airport **LEFT JOIN** `travel-sample` route **ON** airport.faa = route.sourceairport **AND** route.type = "route"
ANSI JOIN with HASH JOIN	SELECT .. FROM `travel-sample` airport **JOIN** `travel-sample` route **USE HASH(**build**)** **ON** airport.faa = route.sourceairport **AND** route.type = "route"
ARRAY JOIN	SELECT .. FROM default b1 **JOIN** default b2 **ON** b2.c21 = b1.c11 AND b2.type = "right" **AND ANY** v **IN** b2.a21 **SATISFIES** v = b1.c12 **END**

N1QL for Query - Example

Movie

```
{
”title": “The Avengers”,
 "casting": [
   {"character": "Tony Stark / Iron Man",
     "name": "Robert Downey Jr",
   },  …
 ],
 "releaseYear": 2019,
 "revenue": 2792269749
}
```

Person

```
{
 "birthYear": 1965,
 "deathYear": null,
 "primaryName": "Robert Downey Jr",
 "primaryProfession":
        "actor, producer, soundtrack"
}
```

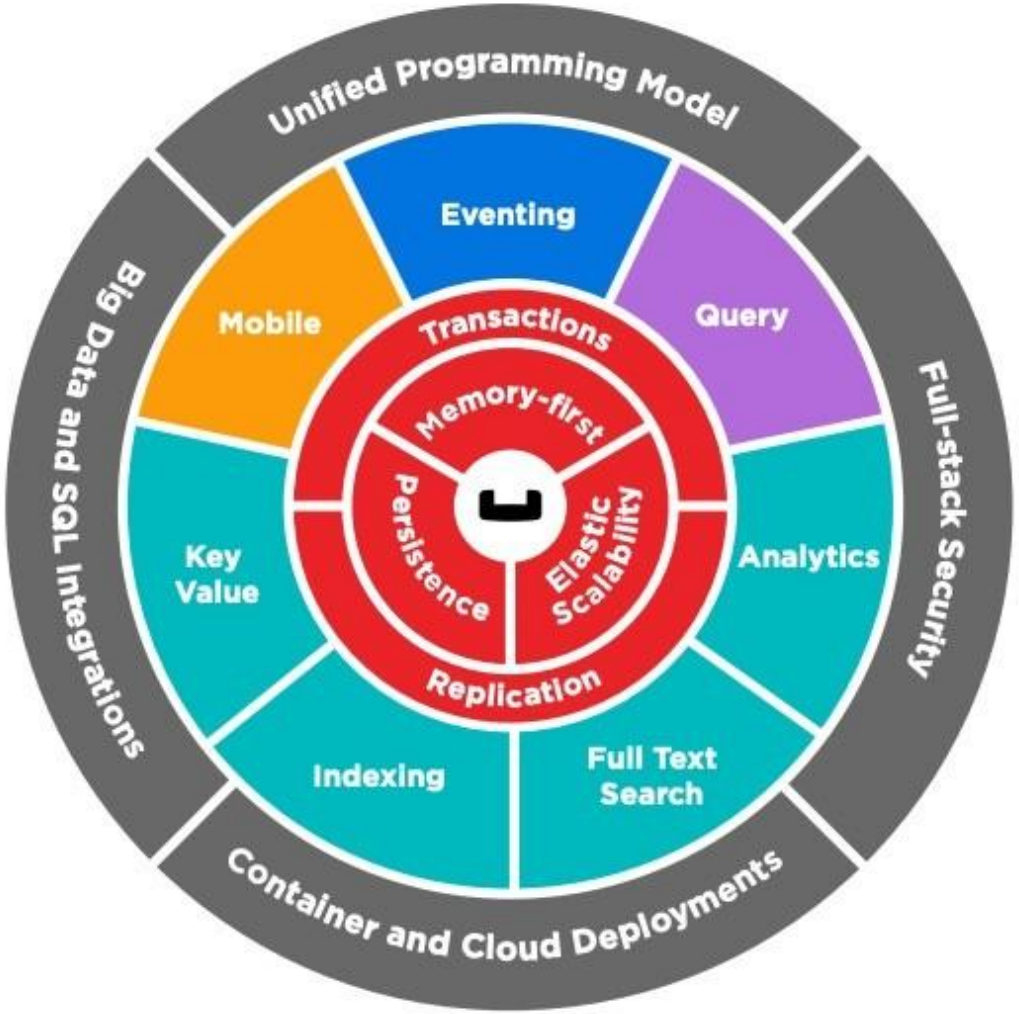
Identify for each movie where Robert Downy Jr. has been playing in who is the youngest actor in the movie, organised by year.

```
SELECT
   MIN(person.birthYear) AS year,
   movie.title
FROM moviedata AS movie
UNNEST movie.casting AS actor
JOIN moviedata AS person
    ON actor.name = person.primaryName
    AND person.type = "person"
WHERE movie.type = "movie"
AND ANY member IN movie.casting SATISFIES
member.name = "Robert Downy Jr" END
GROUP BY movie.title
ORDER BY year DESC
```

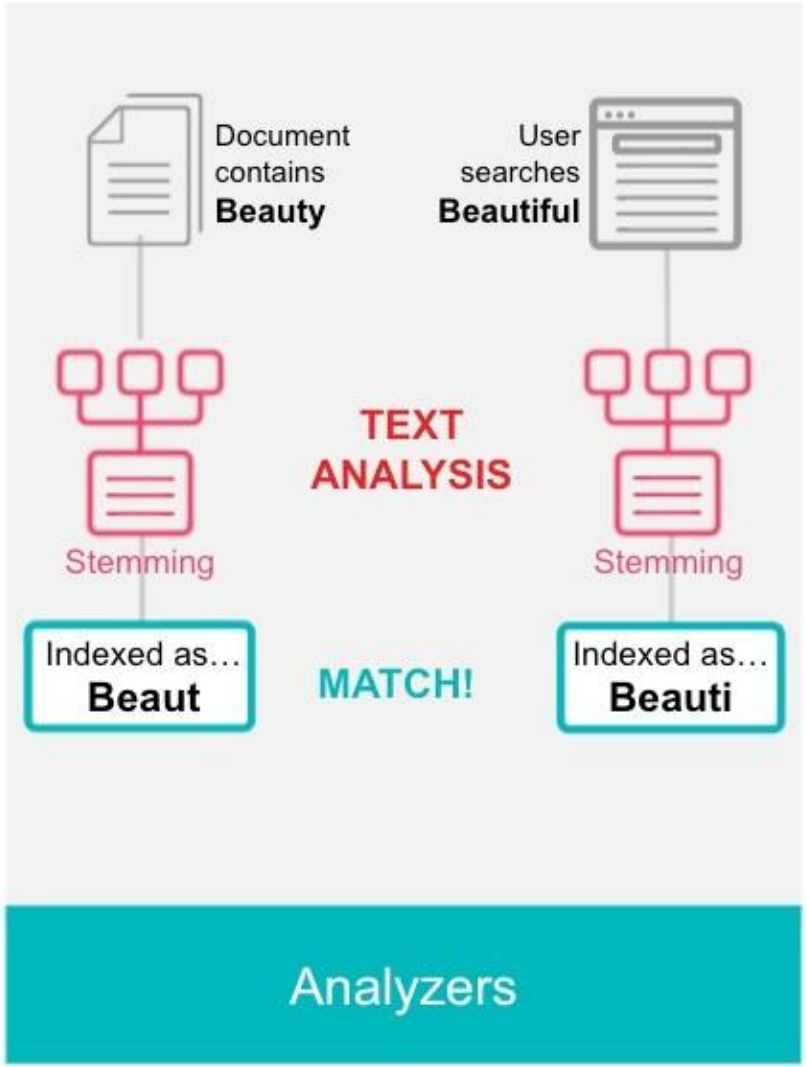
Result

```
[
  {
    "title": "Charlie Bartlett",
    "year": 1978
  },
  {
    "title": "The Avengers",
    "year": 1973
  },
  {
    "title": "Captain America: Civil War",
    "year": 1973
  },
  {
    "title": "A Guide To Recognizing
              Your Saints",
    "year": 1973
  },
 …
]
```

Full Text Search

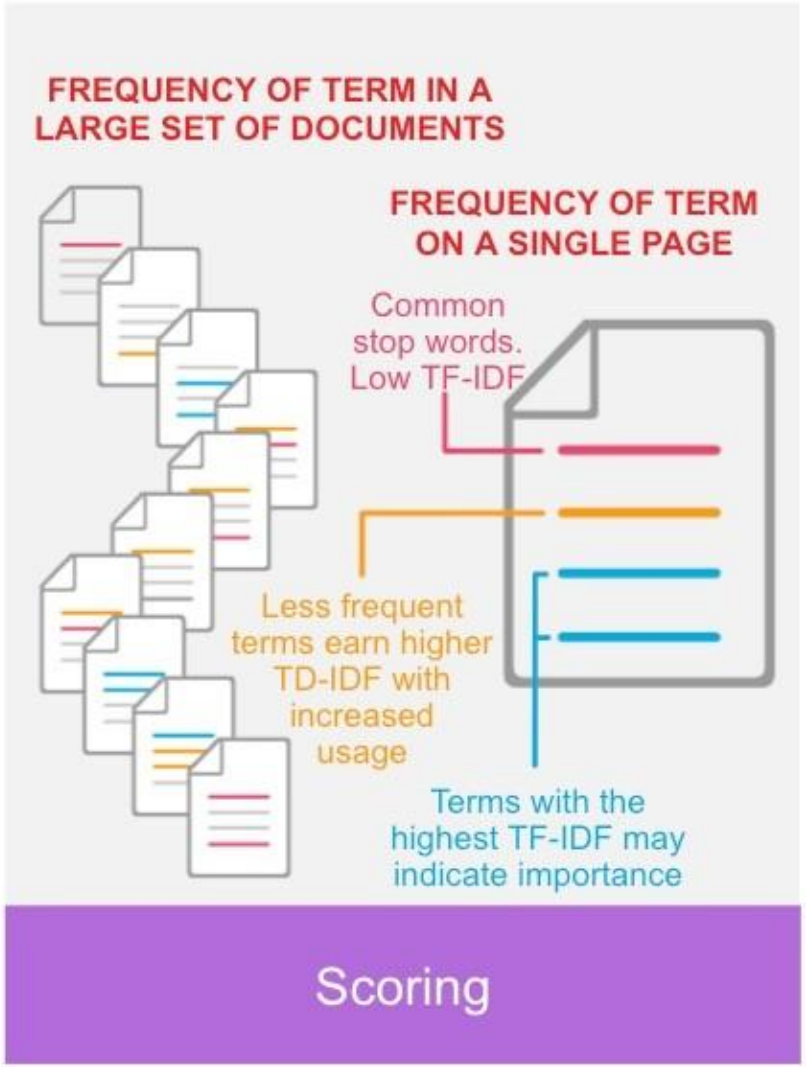


Full Text Search



Analyzers

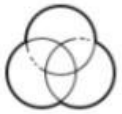
Inverted Indexes



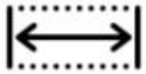
Scoring

Full Text Search

Simple Queries



Compound Queries



Range Queries (string, date, numeric)

String Queries (natural language)

Geospatial Queries

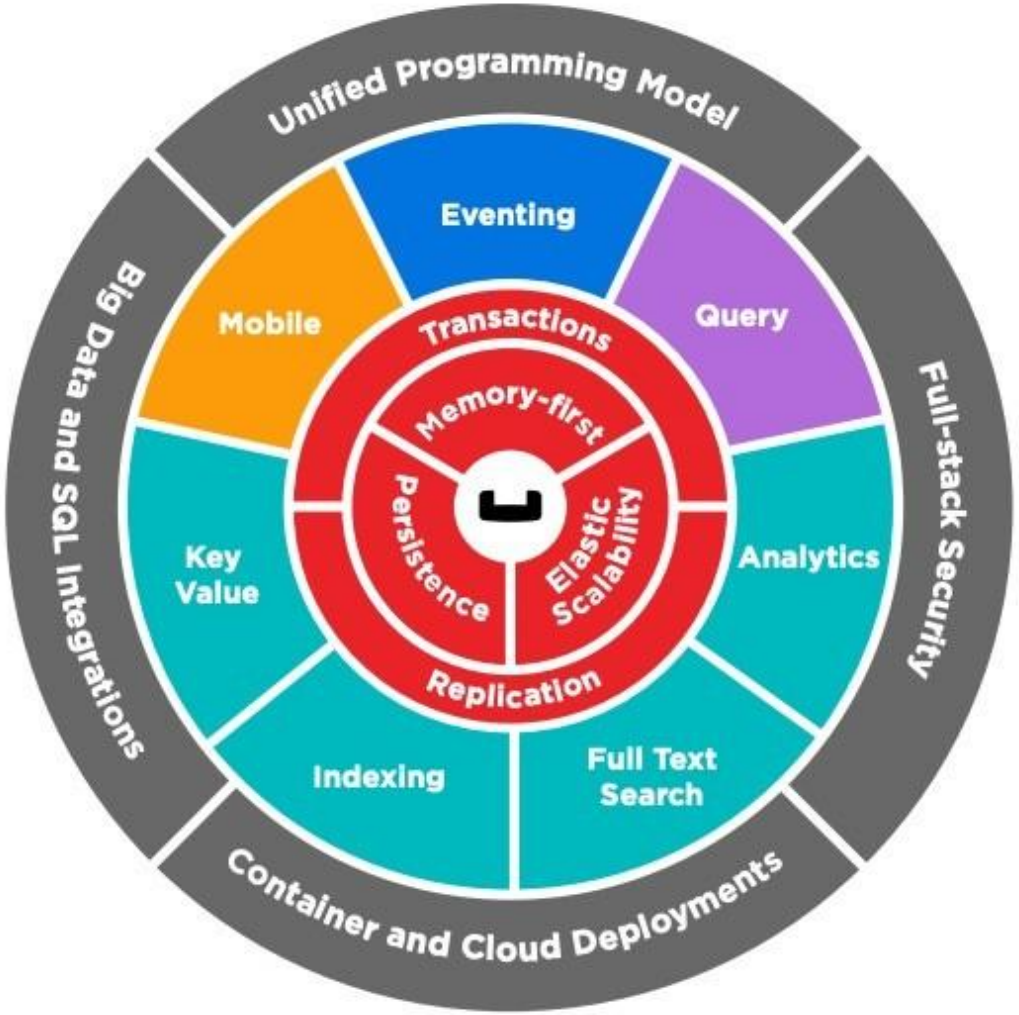
Non-analytic (i.e. exact match)

Special queries (for dev purpose)

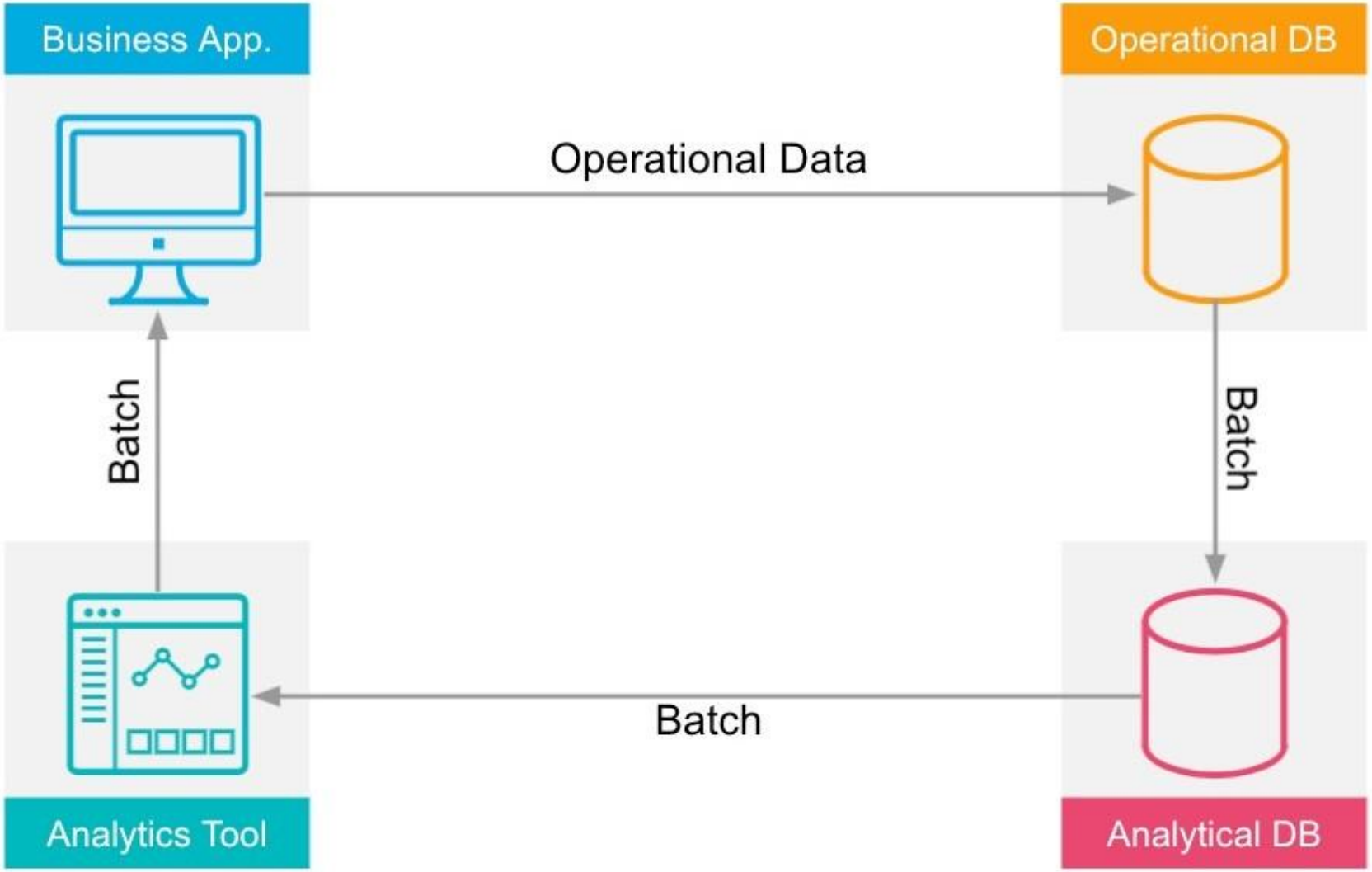
Name	Language
ar	Arabic
bg	Bulgarian
ca	Catalan
cjk	Chinese | Japanese | Korean
ckb	Kurdish
da	Danish
de	German
el	Greek
en	English
es	Spanish (Castilian)
eu	Basque
fa	Persian
fi	Finnish
fr	French
ga	Gaelic
gl	Spanish (Galician)
hi	Hindi
hu	Hungarian

Name	Language
eu	Basque
fa	Persian
fi	Finnish
fr	French
ga	Gaelic
gl	Spanish (Galician)
hi	Hindi
hu	Hungarian
hy	Armenian
id, in	Indonesian
it	Italian
nl	Dutch
no	Norwegian
pt	Portuguese
ro	Romanian
ru	Russian
sv	Swedish
tr	Turkish

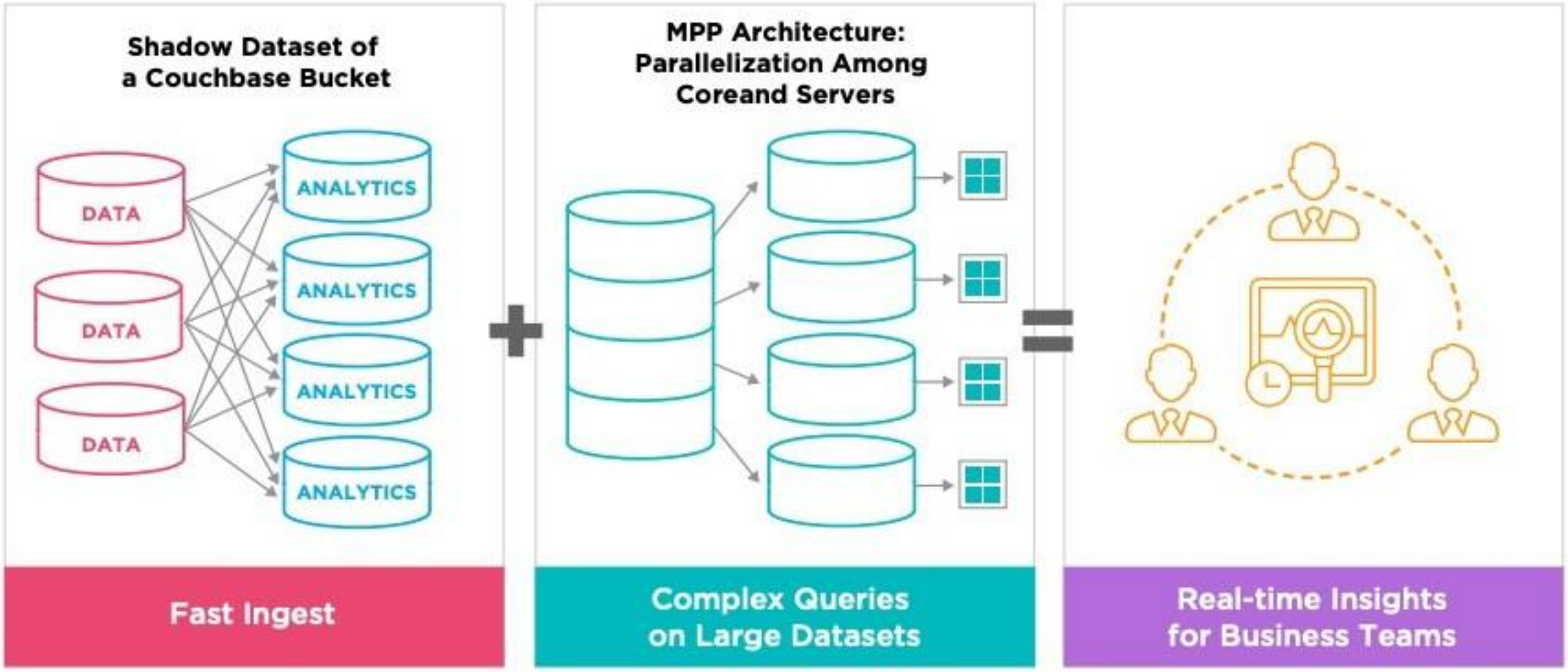
Analytics


Unified Programming Model
Full-stack Security
Container and Cloud Deployments
Big Data and SQL Integrations
Eventing
Query
Analytics
Full Text Search
Indexing
Key Value
Mobile
Transactions
Memory-first
Persistence
Elastic Scalability
Replication

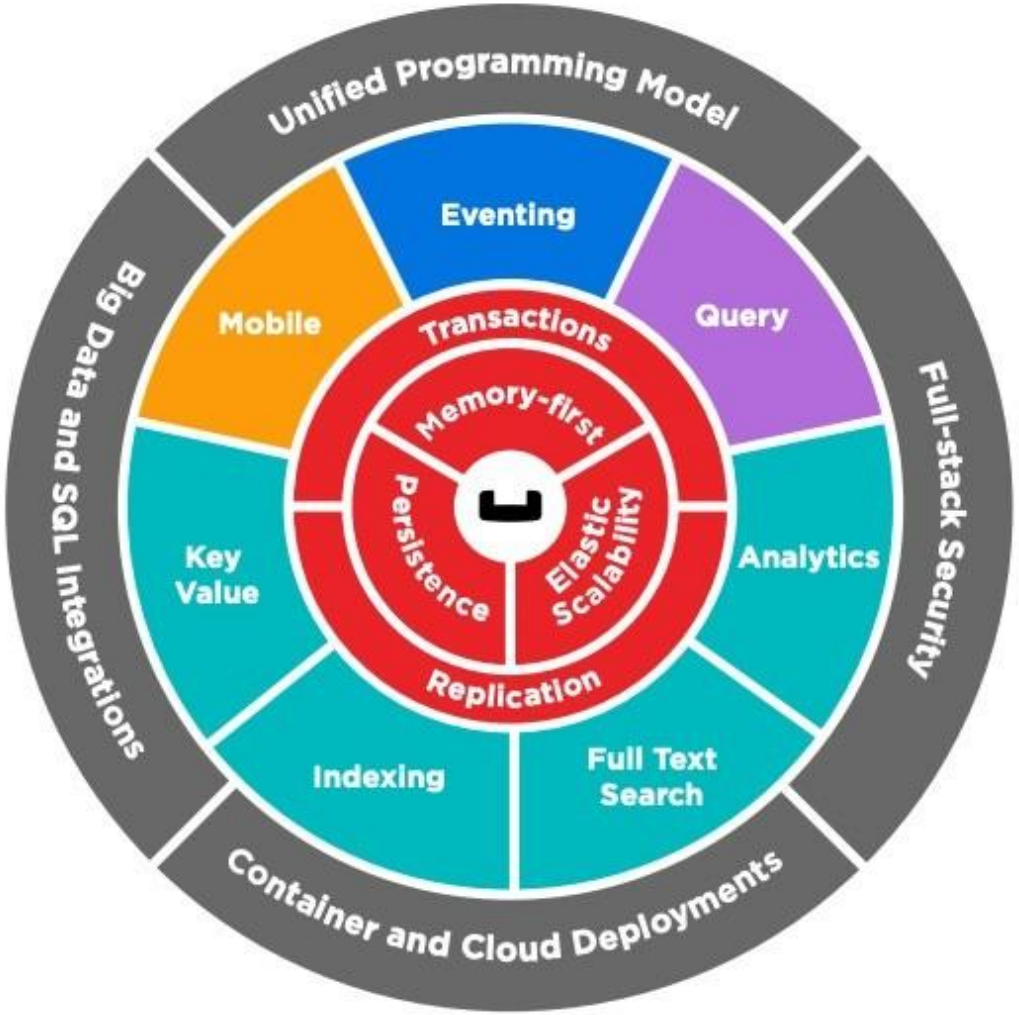
Analytics: A typical setup with ETL



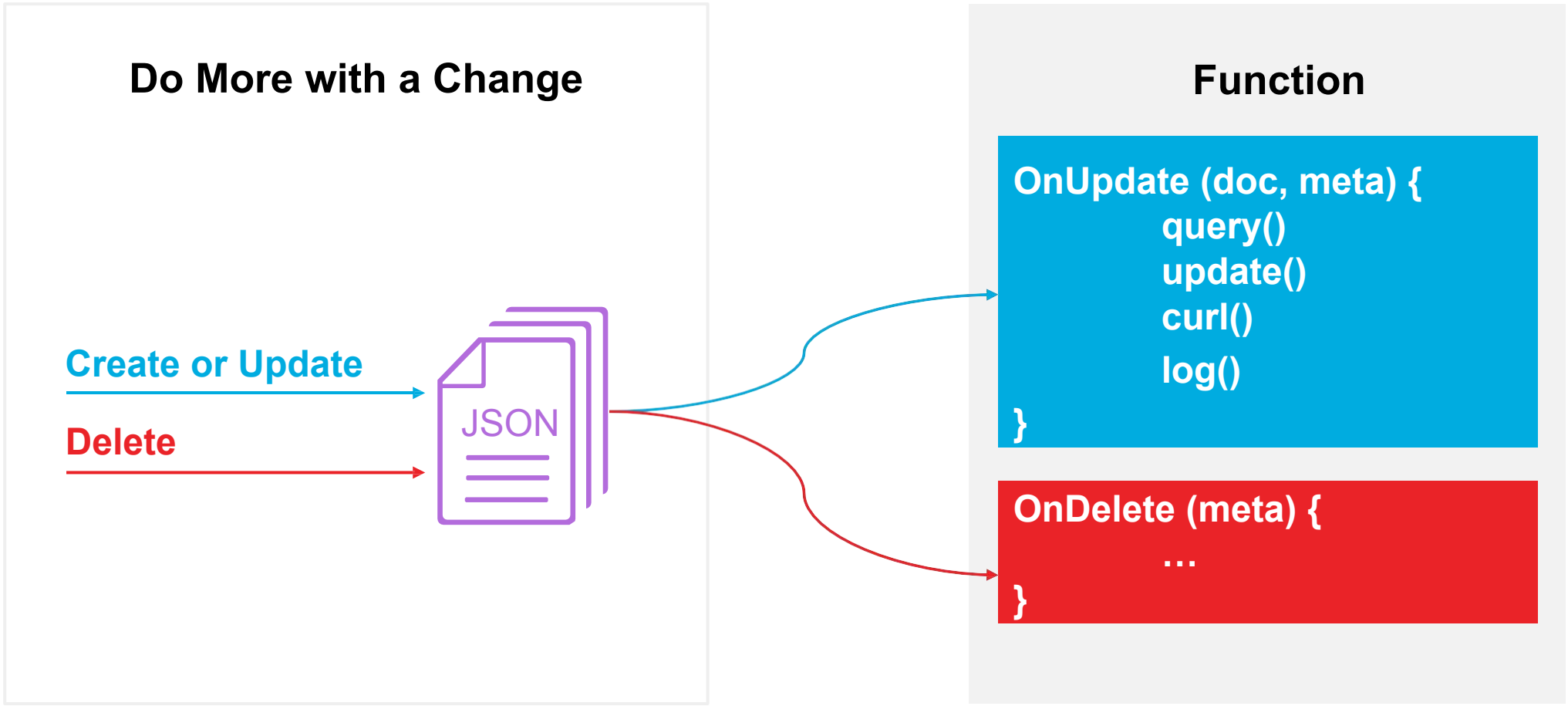
Analytics on Couchbase: NoETL



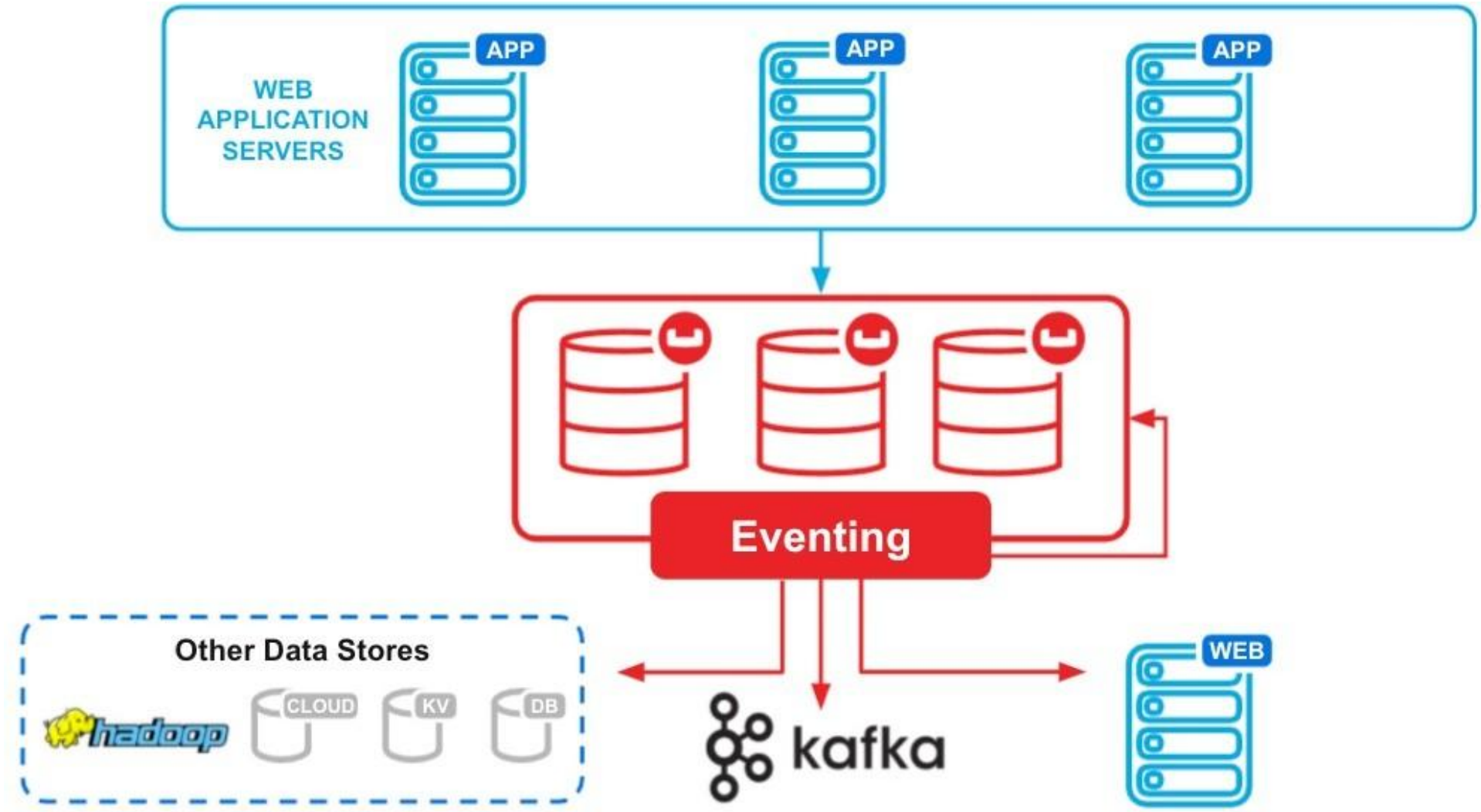
Eventing


Unified Programming Model
Big Data and SQL Integrations
Full-stack Security
Container and Cloud Deployments
Eventing
Mobile
Query
Key Value
Analytics
Indexing
Full Text Search
Transactions
Memory-first
Persistence
Elastic Scalability
Replication

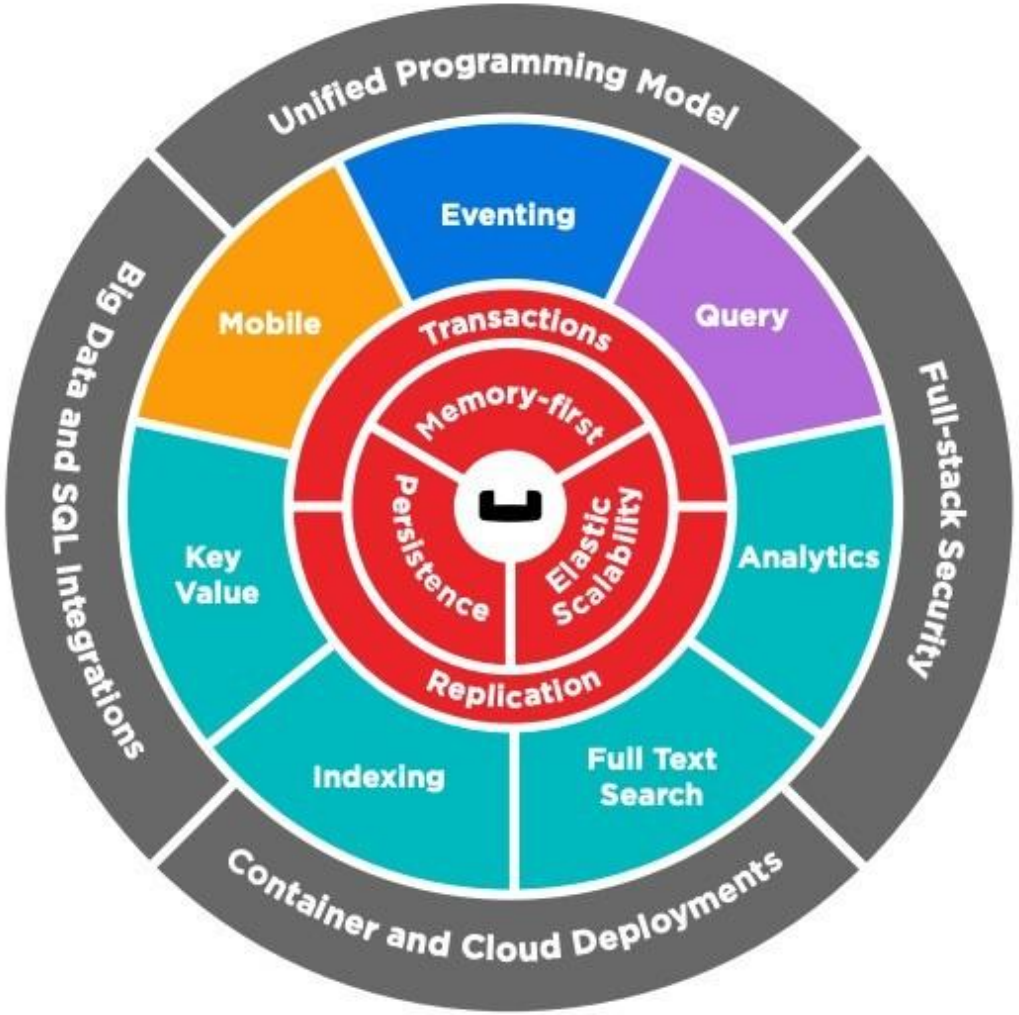
Eventing


Do More with a Change
Create or Update
Delete
JSON
Function
OnUpdate (doc, meta) {
query()
update()
curl()
log()
}
OnDelete (meta) {
…
}

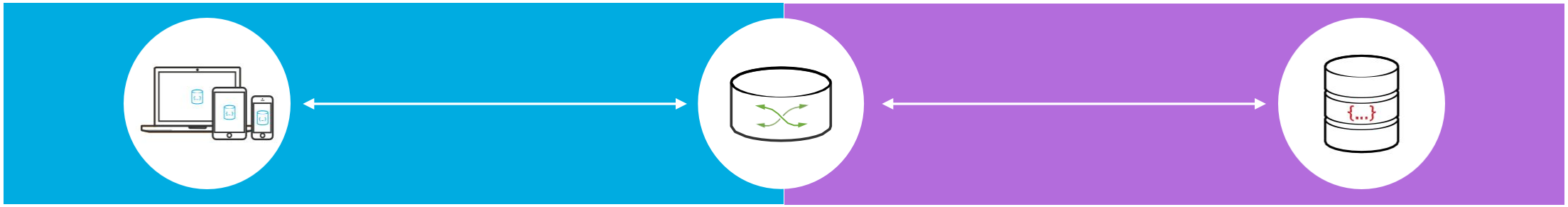
Eventing


WEB
APPLICATION
SERVERS
APP
APP
APP
Eventing
Other Data Stores
hadoop
CLOUD
KV
DB
kafka
WEB

Mobile


Unified Programming Model
Full-stack Security
Container and Cloud Deployments
Big Data and SQL Integrations
Eventing
Query
Analytics
Full Text Search
Indexing
Key Value
Mobile
Transactions
Memory-first
Persistence
Elastic Scalability
Replication

Mobile



COUCHBASE LITE | SYNC GATEWAY | COUCHBASE SERVER

Client | WAN | Middle Tier | LAN | Storage

Lightweight embedded NoSQL database with full CRUD, query and search functionality.

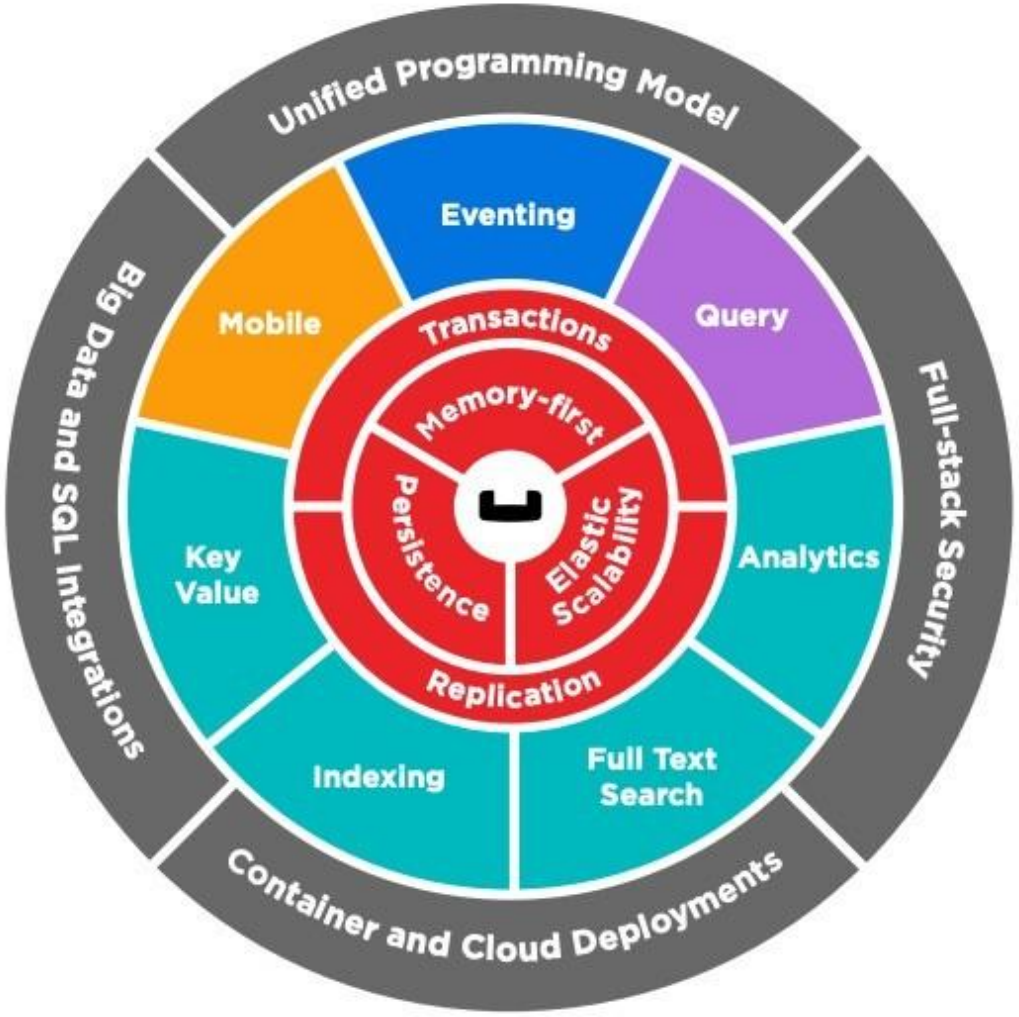
Secure web gateway with synchronization, data access, and data integration APIs for accessing, integrating, and synchronizing data over the web.

Highly scalable, highly available, high performance NoSQL database server.

Security

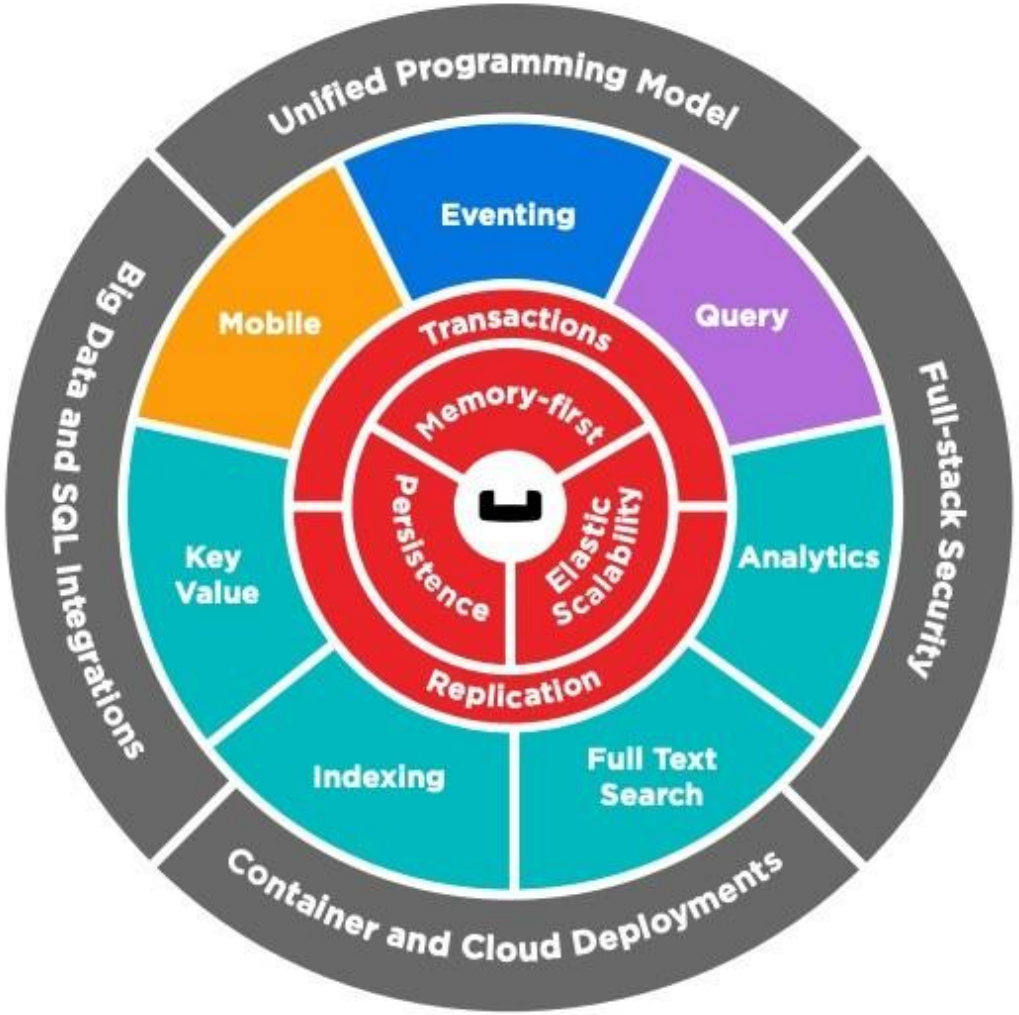
Built-in enterprise level security throughout the entire stack includes user authentication, data access control, secure transport, and full database encryption.

Integrations

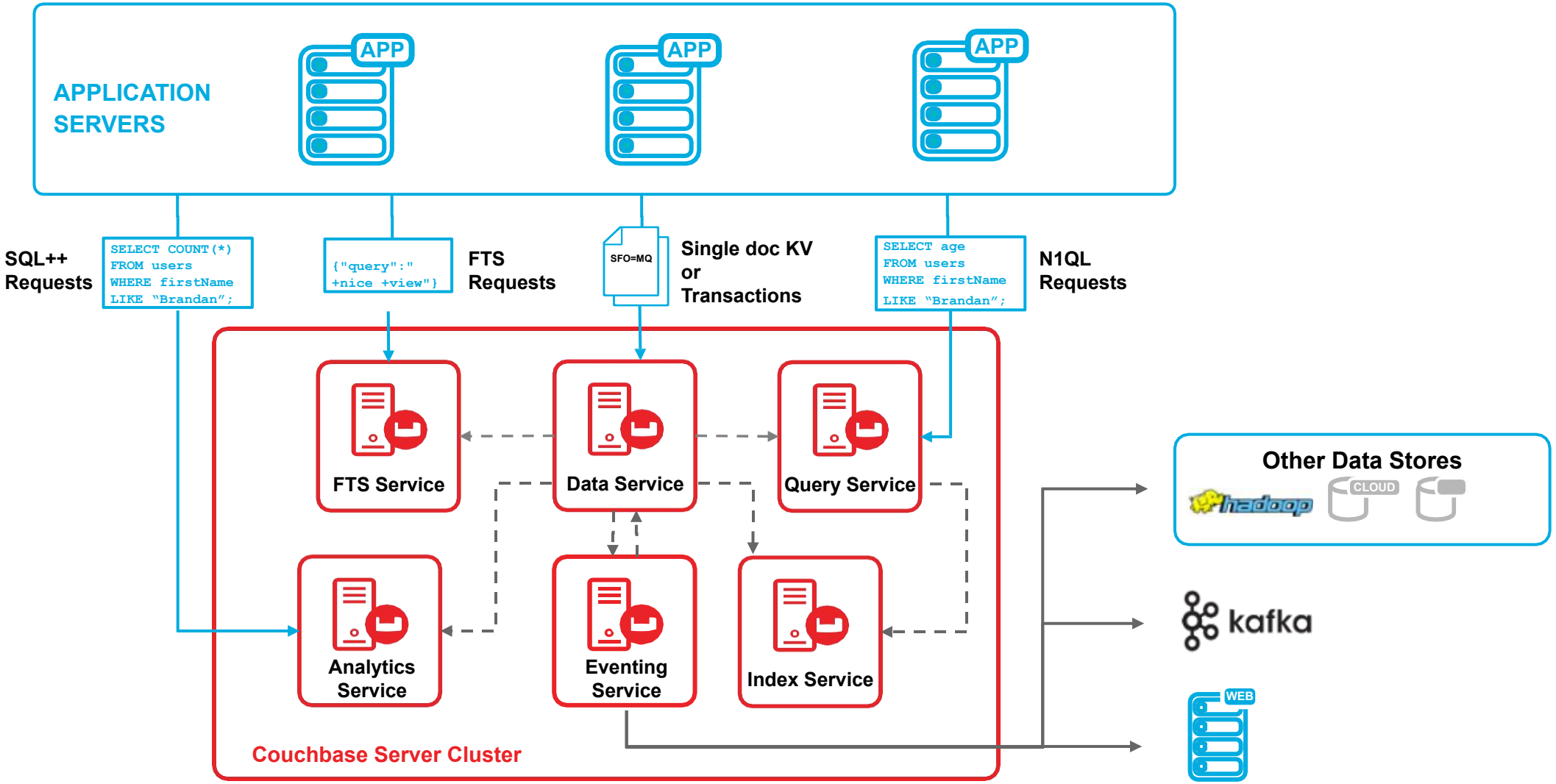


Big Data and SQL Integrations

Unified Programming Model

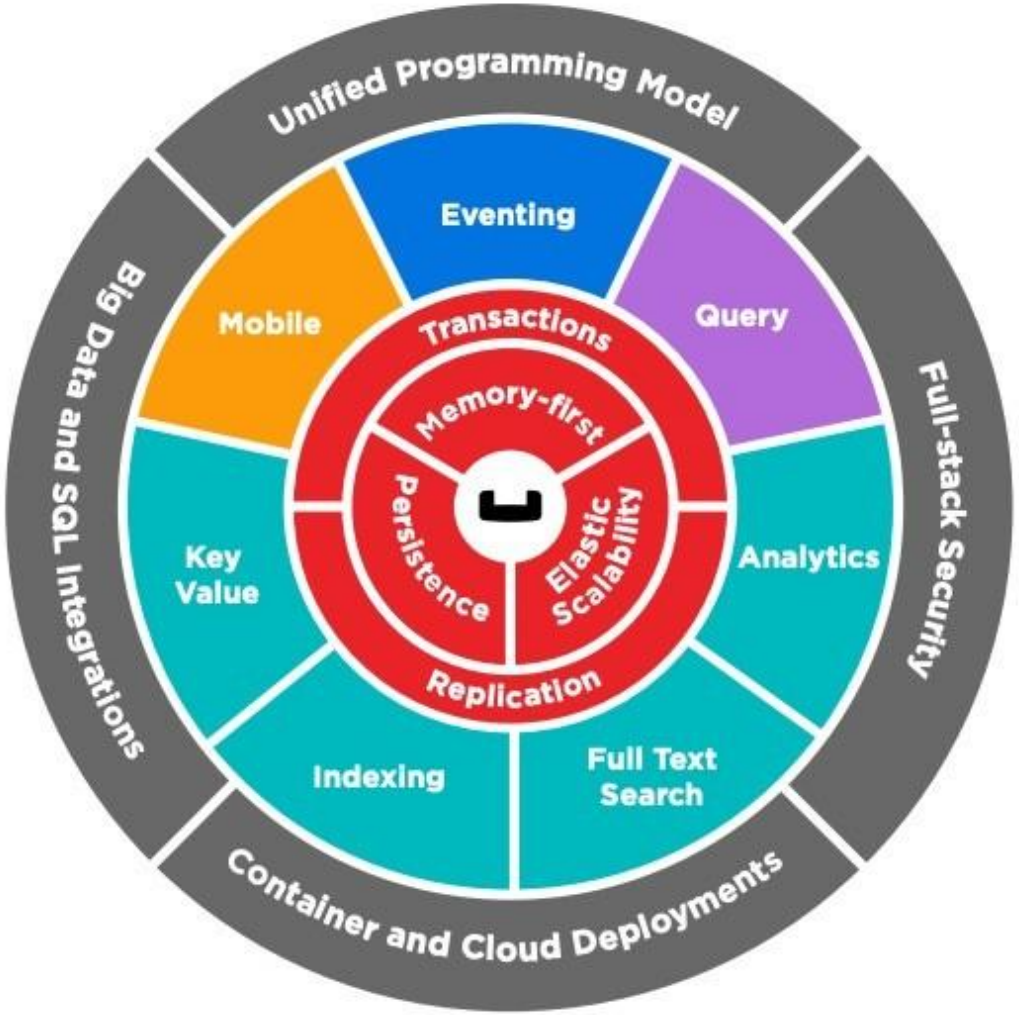


Unified Programming Model – Single API

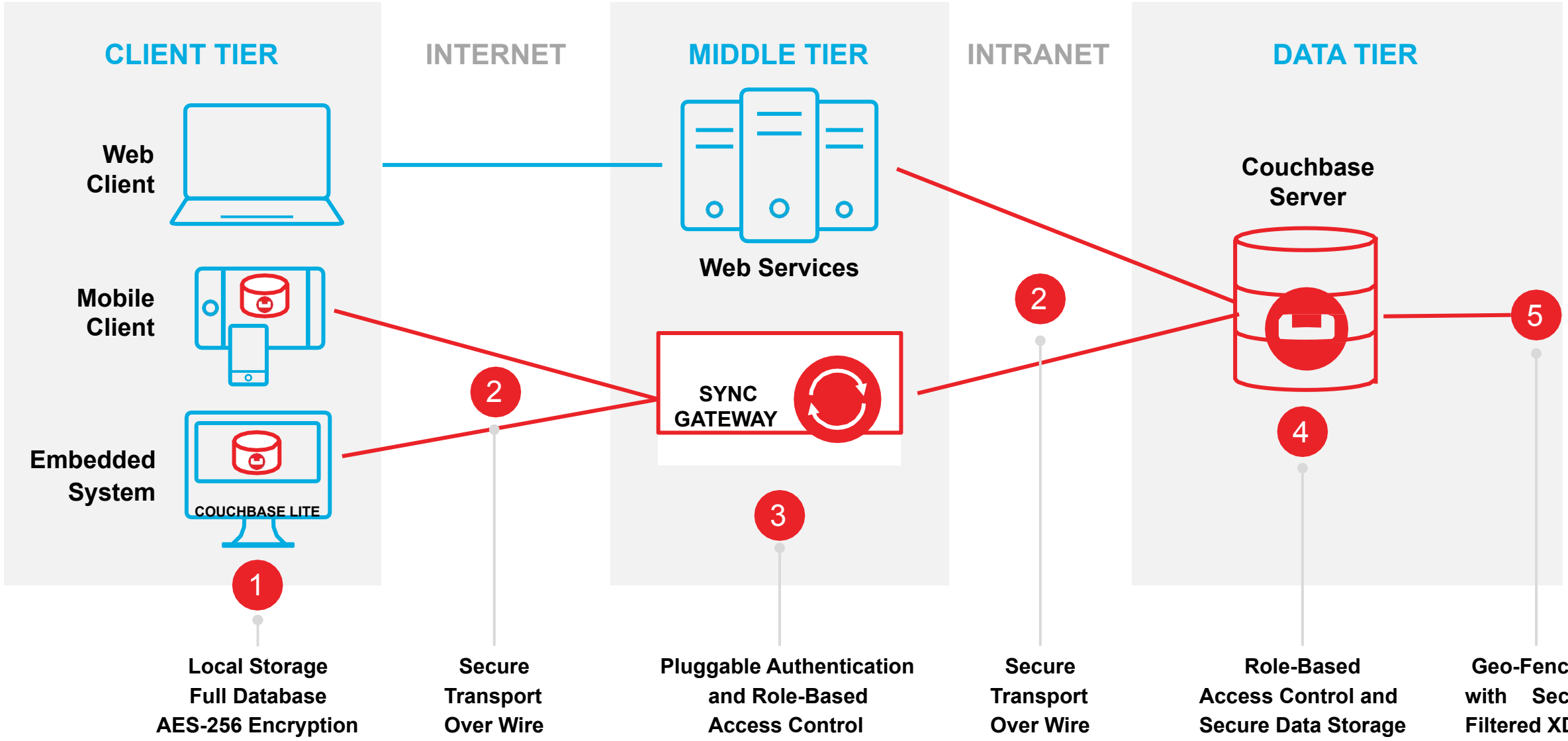


SDKs

Security


Unified Programming Model
Full-stack Security
Container and Cloud Deployments
Big Data and SQL Integrations
Eventing
Mobile
Query
Key Value
Analytics
Indexing
Full Text Search
Transactions
Memory-first
Persistence
Elastic Scalability
Replication

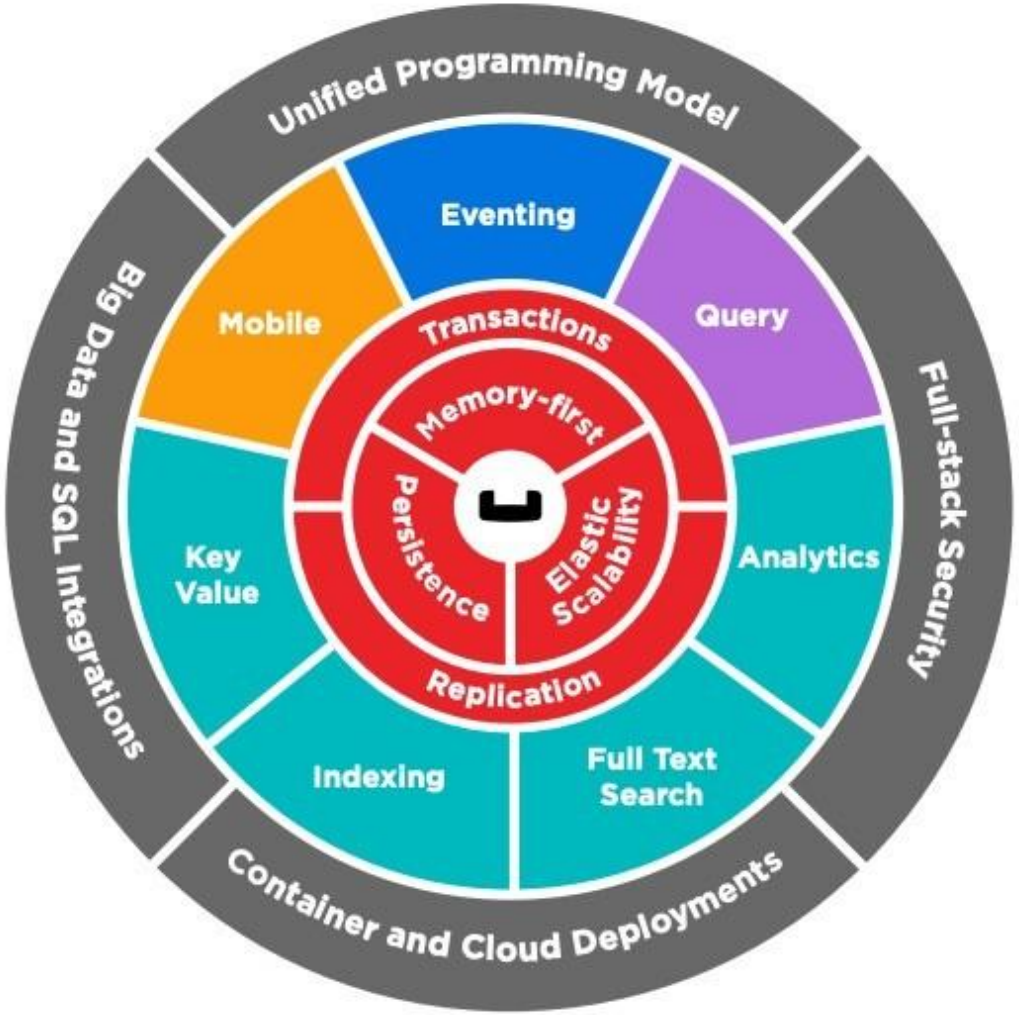
Full Stack Security



Full Stack Security

Authentication	Authorization	Crypto	Auditing	Operations
App/Data: SASL AuthN Admin: Local or LDAP Users or LDAP Groups PAM Authentication	Local Admin User Local Read-Only Admin RBAC for Admins RBAC for Applications (since 5.0)	TLS admin access TLS client-server access Secure XDCR X.509 certificates for TLS Data-at-rest Encryption* Field-level Encryption (since 5.5) Secret Management Support for Configurable TLS Cipher Suites	Admin auditing API request auditing (since 5.5) N1QL auditing (since 5.5)	Security management via UI/CLI/REST

Deployments



Deploy Anywhere

Database-as-a-Service

- Fully-Managed service
- Automated and Self-Service
- Agile and flexible – deploy in minutes
- Highly available, reliable and secure
- Comprehensive monitoring and support
- Eliminate operational overhead

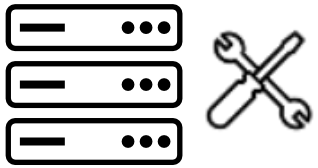
Couchbase Cloud Pu

Self-Managed on IaaS

- Customer managed and operated
- Deployed in the Public Cloud
- Requires DBA's and Infrastructure admins from customer to manage
- Kubernetes Autonomous Operator and deployment templates speed deployment and reduce management burden

ic Clouds

Self-Managed On-Premise



- Customer managed and operated
- Maximum flexibility, customizability and performance
- Requires DBA's and Infrastructure admins to manage
- Kubernetes Autonomous Operator speeds deployment and reduces management burden

Private Clouds

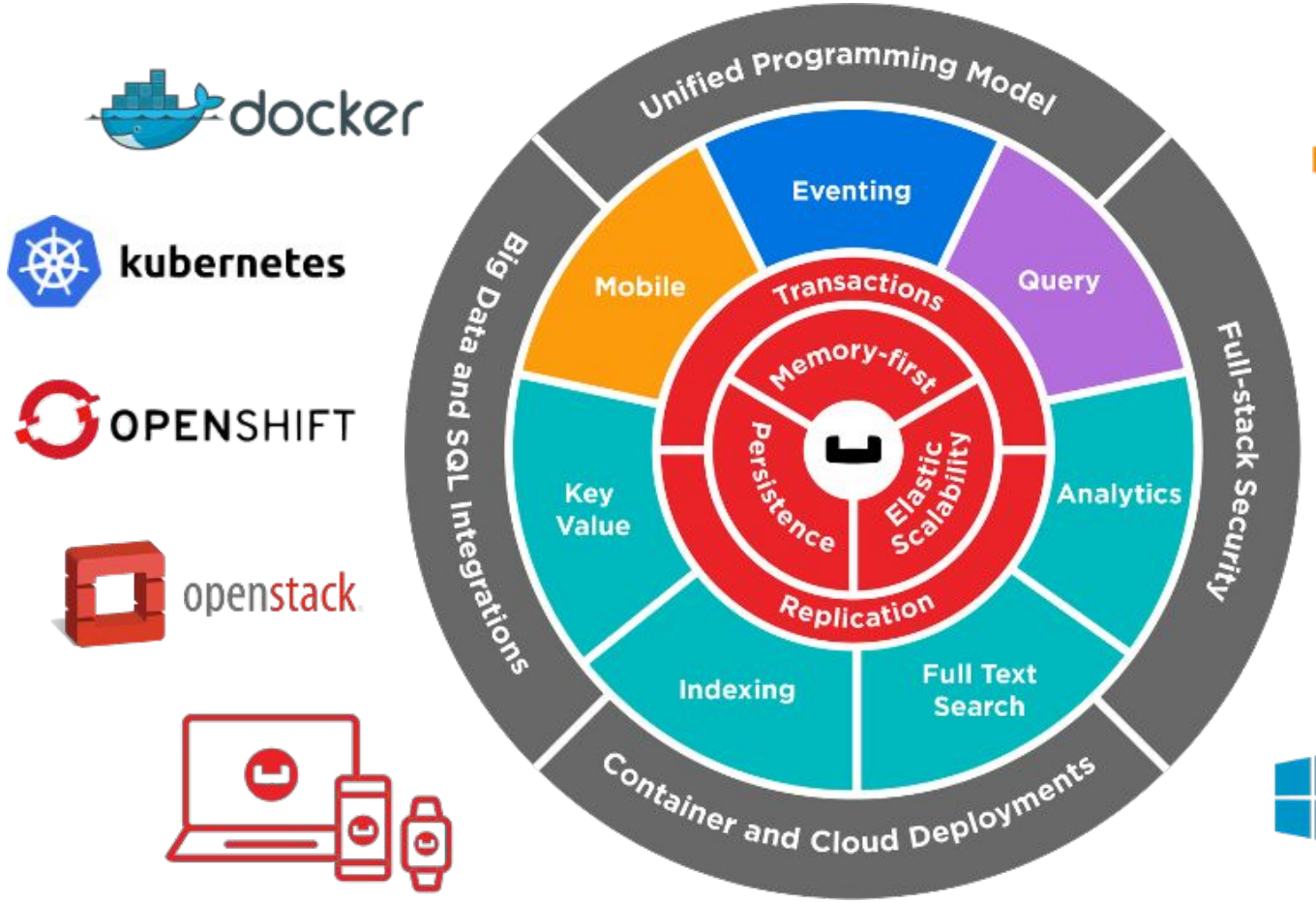
FLEXIBLE DEPLOYMENT OPTIONS TO MEET YOUR REQUIREMENTS

COUCHBASE ARCHITECTURE

ANY CLOUD, ANY DEVICE, MANY SERVICES

ONE data platform

Combines multiple **natively developed services** **Infrastructure-agnostic**



Improving customer experience & engagement

Faster innovation & time to market

Reducing infrastructure & operations costs

Reducing risks

Схема лицензирования

https://www.couchbase.com/products/editions

Вопросы?

Практика

ДЗ

ДЗ

Развернуть кластер Couchbase

Создать БД, наполнить небольшими тестовыми данными

Проверить отказоустойчивость

Рефлексия

Рефлексия

Как вам couchbase?

И так как нет доступа в Azure & AWS заменяем на ЯО & SberCloud

также расшарю предыдущие лекции, если кому актуально

Заполните, пожалуйста,
опрос о занятии по ссылке в чате
https://otus.ru/polls/52770/

Спасибо за внимание!
Приходите на следующие вебинары

Аристов Евгений