



# ARCHITECTURE GUIDE

## CrateDB Key Concepts

January, 2024

# Table of Contents

## [Architecture Overview](#)

### [Open Source Licensing Model](#)

### [Multi-Model Database](#)

- Document/JSON
- Relational Data
- Full-Text Search & Search Engine
- Vector Store and Similarity Search
- Time Series Data
- BLOB Data

### [Native SQL Syntax](#)

### [Distributed Shared-Nothing](#)

#### [Architecture](#)

- Node Architecture
- Cluster State Management
- Horizontal Scalability
- High Availability

### [Data Storage](#)

- Partitioning and Sharding
- Replication
- Atomicity, Durability, and Consistency
- Advanced Indexing
- Columnar Storage
- Data Tiering: Hot, Warm, and Cold Data

### [Performance: High-Volume Concurrent Reads and Writes](#)

- Distributed Query Engine
- Distributed Writes
- Distributed Reads

## [Security](#)

- Data encryption
- Authorization
- Auditing
- ISO 27001 Certification

## [Flexible Deployment Models](#)

- Containerized
- Private and Public Cloud
- On-Prem & Hybrid
- Edge
- Logical Replication Between Clusters
- Zone Aware Deployments

## [Ecosystem Integrations](#)

- Programming Languages
- SQL- and API based Integrations
- Data Ingestion & Streaming
- Data Processing, ETL and Orchestration
- Data Visualization & Analytics
- Monitoring & Logging Tools
- AI and Machine Learning Applications

## [Getting Started with CrateDB](#)

- CrateDB Cloud & Free Tier
  - Community & Learning Resources
  - Speak to a Technical Expert
- 

# Architecture Overview

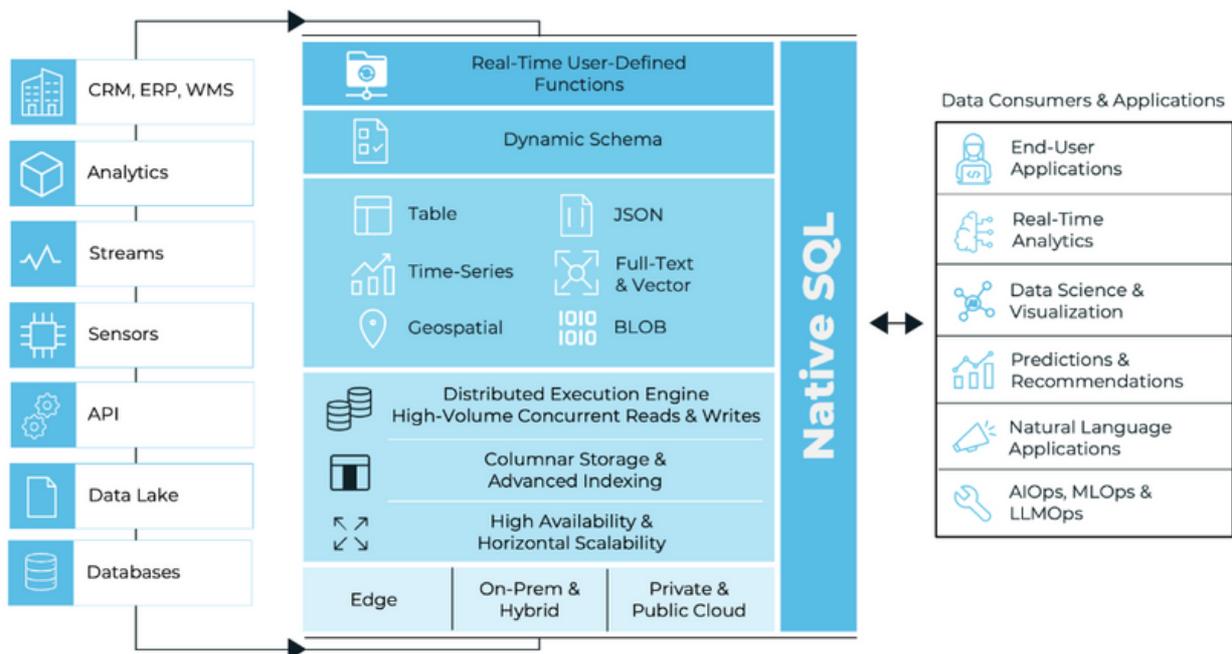
CrateDB allows you to store any kind of **structured**, **semi-structured** and **unstructured** data in one single technology. It also offers a **dynamic schema** for rapid evolution and provides access to data via **SQL**. It streamlines data management by blending operational and analytical data to enable modern application development. It offers the best of SQL, NoSQL, and search engines and enables rich full-text and vector similarity search to power both traditional and AI-enabled applications, ranging from real-time analytics over machine learning, to natural language processing and LLMs.

Its distributed, shared-nothing architecture enables **high availability**, **vertical and horizontal scaling** as well as **high-volume concurrent reads and writes**.

CrateDB does not just provide faster time-to-market, but also very low operational overhead. **Scaling** is a matter of adding nodes, the database itself takes care of data distribution.

The combination of advanced indexing and columnar storage enables **very fast** queries in single-digit milliseconds across billions of rows.

The **flexible deployment options** provide the level of flexibility needed for any use case: as a fully managed service on AWS, Azure and GCP; as a self-deployed solution in your own cloud environment or on-premises; as a solution deployed at the Edge.



CrateDB Architecture Overview

# Open Source Licensing Model

CrateDB embraces and supports the open source development model. Its source code is available on GitHub under the Apache 2.0 License, allowing users to access and contribute to its ongoing development.

On one hand, CrateDB offers the flexibility of self-deployment, enabling organizations to customize and tailor the database to their specific needs.

On the other hand, users can access CrateDB as a service through the CrateDB Cloud offering, simplifying the setup and management process.

For all deployment models, support subscriptions are available on demand, with “Basic” and “Premium” options.

CrateDB Cloud paid plans include “Basic” support by default.



# Multi-Model Database

CrateDB is a **multi-model database**. Its strength lies in the **efficient handling of multiple data models within the same database and even within the same table**:

- structured data (tables, time series, geospatial)
- semi-structured data (JSON / documents)
- unstructured data (text, vector, binaries like documents, images, and videos)

It eliminates the need to manage and synchronize multiple database technologies and learn different languages by offering unified access via the well-known SQL language.

- All data models can be combined in the same record.
- All data models are **accessible via SQL**, the well-known query language, allowing for complex queries, full-text and vector search.
- Complex objects and nested objects can be stored with no human intervention. Data can be directly inserted as a JSON string.
- New columns, supporting any data type and format, can be dynamically added, without table locks, allowing for seamless adaptation to changing needs and requirements. An important CrateDB's strength lies in its schema flexibility.
- CrateDB addresses the traditional challenge of the rigidity inherent in relational schemas and the complexity associated with making changes in a production environment.
- Columnar storage enables fast aggregation across individual records, making it suitable for combined operational and analytical use cases.
- Binary large objects (BLOBs) can be stored separately from the main database workload, reducing storage costs, with streamlined read/write access via the HTTP REST API.

CrateDB can work with data from various sources – enterprise application data (like CRM, ERP), analytics data, streams data, sensor data, APIs, and data from data lakes or other databases – and can accept many formats with no need for complex data transformations.

