

## • Overview

Cloud2DB is a universal database for Big Data computing. While there are many Big Data database offerings geared towards better performance and scalability, these lack the built-in support for standards, structure and interoperability that is required to effectively and quickly implement enterprise class applications. This is where Cloud2DB excels. Cloud2DB provides a standards-based abstraction layer over these Big Data technologies to provide you with performance and scalability of Big Data database platforms along with structure, standards and interoperability of relational database platforms.

## • Interoperable

Cloud2DB is a “universal database” that offers interoperability with existing tools and technologies within your company’s current environment. Further, Cloud2DB works seamlessly with most of the popular Big Data database technologies like Cassandra, MongoDB, DynamoDB, Redis, Google Cloud (Bigtable, Spanner, Firestore, Memorystore). It also supports the relational database standards ANSI SQL-92/ANSI SQL-99 and JDBC 3.0.

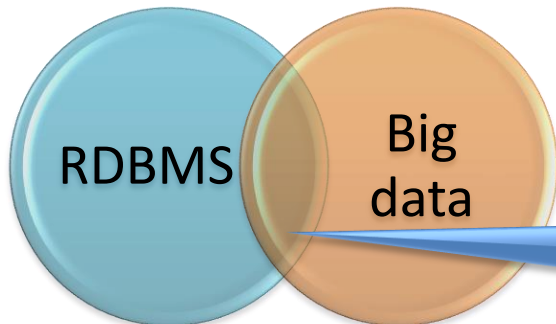
## • Scalable

Cloud2DB brings relational database functionality to Big Data computing platforms by bridging the gap between Big Data database technologies and relational database technologies. Whether you have a need to bring the scalability and performance of the Big Data platforms (like Cassandra, MongoDB) in-house, or plan on using SaaS based Big Data platforms (like Amazon DynamoDB or Google Bigtable), Cloud2DB is the right solution for you.

## • No Lock In

With the flexibility offered by Cloud2DB, you no longer have to fear about being locked into a new emerging technologies. You have an option to revert back to your current environment with ease, as well as change to another underlying Big data platform.

## • Cloud2DB Model



## • Customer Pain Points

Most of the organizations use relational database systems as their database platforms. These organizations are very interested in exploring Big Data databases as another option for their enterprise database platforms.

However, today’s Big Data database platforms are proprietary in nature and require complete re-engineering of existing applications to use them as enterprise database platform.

As such, organizations face the dilemma of effectively utilizing their existing tools, technologies, skill sets and methodologies employed in their organization, while migrating to more efficient computing environments.

Cloud2DB addresses these critical issues.

### Traditional RDBMS

Examples: Oracle, Sybase, DB2

Pros:

- Easy to scale up, predictable performance, standards based (SQL and JDBC) and Interoperable

Cons:

- Cannot scale out (lacks elasticity), time-to-Market can be long, high price limits accessibility by many companies

### Open Source RDBMS

Examples: MySQL, Postgre

Pros:

- Low price; Free – price point provides high accessibility

Cons:

- Cannot scale up or out (lacks elasticity)
- Support needs to be purchased from a third party

### Pure Big data Databases

Examples: Cassandra, MongoDB, DynamoDB, Redis, Google Cloud (Bigtable, Spanner, Firestore, Memorystore)

Pros:

- Easy to scale up and scale out (elastic)
- Per use price or free

Cons:

- Limited out-of-the box interoperability with existing tools (e.g., report writers) so that customized tools have to be built and existing skill sets cannot be easily redeployed

Cloud2DB brings relational database functionality to Big Data computing platforms by bridging the gap between Big data database technologies and relational database technologies.

## • Cloud2DB Solution

### Quick to Market

- Cloud2DB is **PLUG-AND-PLAY**: Allows for an immediate adaption of Big Data database technologies into your enterprise technology stack. This is truly a plug-and-play into the Big Data database technologies.

### Scalability

- Cloud2DB is **SCALABLE**: Provides performance and scalability of Big data database platforms.

### Interoperable

- Cloud2DB offers **STANDARDS, STRUCTURE AND INTEROPERABILITY**: Provides a standards-based abstraction layer over Big Data database technologies to enable interoperability.

### Agile

- Cloud2DB is **ADAPTABLE**: Works seamlessly with existing tools, Technologies, software frameworks used in the organization.

### Simple

- Cloud2DB is **SIMPLE**: Provides JDBC 3.0 and SQL-92 Interface to Big Data database platforms which allows an application developer with basic programming skills to implement business applications on Big Data database platforms.

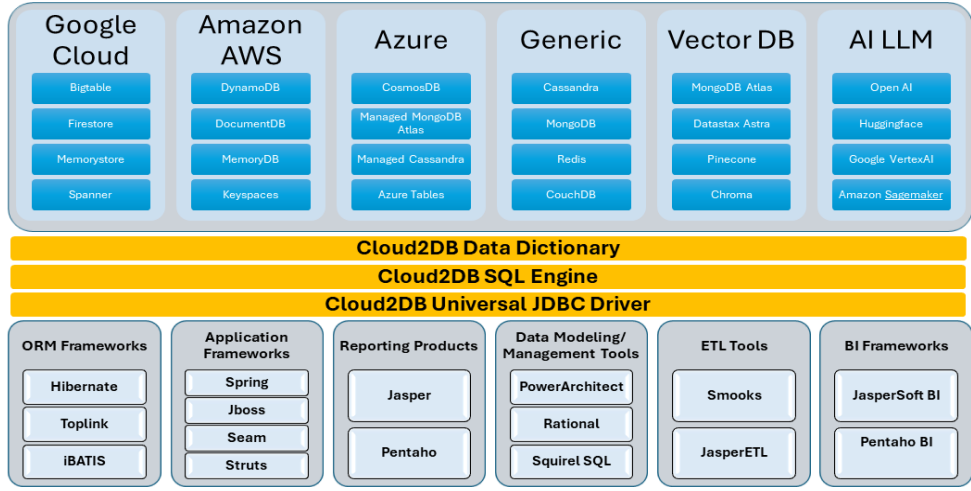
### Streamlined

- Cloud2DB is **NON-DISRUPTIVE**: Any business application written using JDBC and SQL can be ported to Big Data database platforms with little changes in the application code.

**Cloud2DB Functionality**

- Big data database support
  - Cassandra
  - MongoDB
  - DynamoDB
  - Redis
  - Google Cloud
  - Amazon AWS
  - Azure
- Tools support
  - All JDBC compliant tools
  - Squirrel SQL (database management)
  - Power Architect (data modeling)
  - Jasper Reports (reporting)
- Framework support
  - All JDBC compliant frameworks
  - Hibernate (OR mapping)
  - Spring (Data access from application)
- ANSI SQL-92/ANSI SQL-99 support
- Referential integrity (Primary Keys, Foreign Keys)
- Role Based Security
- Joins (Inner Join, Left Outer Join, Theta Join, Cross Join)
- Sub-queries (Exists, Not Exists, In)
- DDL & DML
- Transactions
- ANSI SQL stored functions
- Views
- JDBC 3.0 support

**How Cloud2DB Works**



The Bigdata databases mentioned in the top layer are just representative databases. Cloud2DB framework allows you to expose any Bigdata database via Relational, SQL and JDBC standards.

The products mentioned in the bottom layer are just representative products. Cloud2DB works with any JDBC compliant product.

**Value Proposition – Database Engine Layer**

Today	Cloud2DB
<ul style="list-style-type: none"> <li>Various Big Data database platforms are available</li> <li>Technology: Proprietary to each platform</li> <li>Different skill-sets required for designing for different Big data platforms</li> <li>Separate skill-sets required than commonly used relational database technologies</li> <li>Business change is expensive and requires re-engineering because of proprietary nature of Big Data platforms</li> </ul>	<ul style="list-style-type: none"> <li>Universal database engine works transparently with various Big data database platforms</li> <li>Exposes various Big data database platforms via standard JDBC and SQL interfaces</li> <li>Eliminates need for learning new skills to implement applications on Big Data database platforms</li> <li>Eliminates the need for using non-standard tools to manage Big data databases</li> <li>Makes your existing relational database instances completely portable across various Big Data database platforms</li> </ul>

**Value Proposition – Application Layer**

Today	Cloud2DB
<ul style="list-style-type: none"> <li>Various Big data database client APIs are available</li> <li>Technology: Proprietary APIs to each platform</li> <li>Special and expensive skill-set required for writing applications for Big data databases</li> <li>Business change is expensive and requires re-engineering because of proprietary nature of Big Data platform APIs</li> </ul>	<ul style="list-style-type: none"> <li>Exposes various Big data database platforms via standard JDBC and SQL interfaces</li> <li>Eliminates need for learning new skills to implement applications on Big Data database platforms</li> <li>Eliminates the need for using non-standard tools to manage Big data databases</li> <li>Makes your existing applications completely portable across various Big Data database platforms</li> </ul>

**Value Proposition – Integration Layer**

Today	Cloud2DB
<ul style="list-style-type: none"> <li>Technology: Home grown solutions</li> <li>Home grown solutions are arbitrary and difficult to write</li> <li>Home grown solutions are expensive because of development and maintenance costs</li> </ul>	<ul style="list-style-type: none"> <li>Exposes various Big data database platforms via standard JDBC and SQL interfaces thus making integration almost a non-issue</li> </ul>