

# Modernizing Legacy Data Platforms

TiDB provides lower maintenance costs, simplified architecture, and rich query capabilities at scale through the modernization of legacy systems.

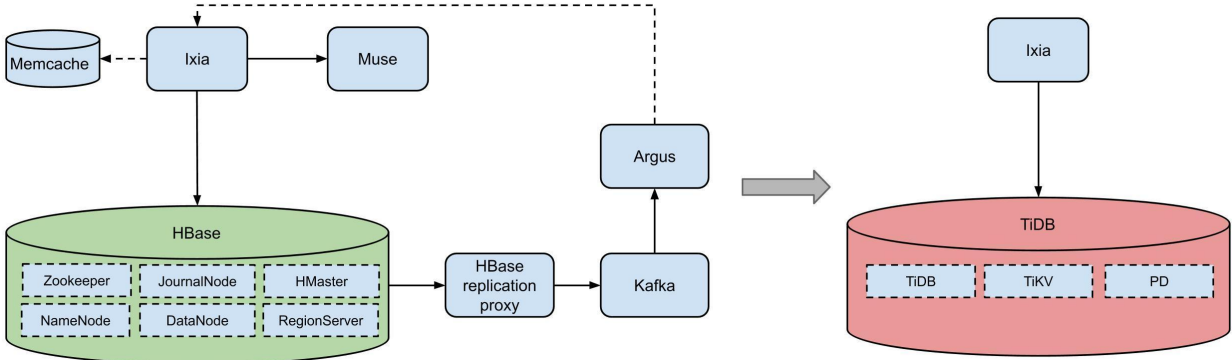
## Key Benefits

TiDB delivers a modern database environment, developed from the ground up to use commodity hardware, provide scalability and data availability while delivering excellent query performance. Some benefits of moving to a modern data platform include:

- Improved developer velocity to meet diverse user requirements
- Reduced costs since you are consolidating a stack of disparate systems
- Enhanced analytics capabilities from a single database
- Multi-cloud availability, enabling you to spread the risk and lessen vendor lock-in

## Example Architecture

The diagram below illustrates a typical customer use case, with data inputs and outputs with TiDB in the middle processing the data for the specific need.



Of note is the fact that many more traditional products, like HBase, Muse, and Kafka, are still in use with TiDB. While it is possible to move to a completely new solution, it is not a requirement for TiDB. Since TiDB is MySQL compatible, the migration can often be accomplished with minimal, or no, changes needed to your existing applications. Developers keep working with the tools they know and users continue to use the front-end tools that they are already using.

With lower need to retrofit the entire environment, implementing TiDB is far less daunting than may be initially believed. Within the TiDB environment, you can implement just the key-value

store, which provides enhanced transactional functionality, or you can also add the column store to benefit from improved performance on analytical queries.

Regardless of which you choose, TiDB manages the storage and optimizes each query to determine how it is best processed. If you use both the row and column stores, the query optimizer determines which data store will provide the best response time and automatically directs the query to that store. Your users do not need to change their work processes, but they will receive much faster response times.

If you are looking for a high-end transactional database, TiDB fills that need as well. Built-in high availability ensures that your data is available and financial grade security keeps it safe from prying eyes. Because TiDB separates the compute and storage layers, each can be scaled independently of the other, providing you with an environment that is consistently right-sized.

## Example Applications

Reasons for migrating from NoSQL to TiDB include the ability to achieve stronger consistency, lower maintenance cost, good performance while maintaining high availability and easy scalability. TiDB delivers a system that supports ACID transactions, secondary indexes, rich query capabilities to handle business growth with ease, and provides an easy path to add real-time analytics to your database environment.

You can meet diversified user requirements with rich query capabilities, distributed query processing, and strong data consistency. With distributed query processing, complex queries can be served with fast responses. Distributed transactions mean that you do not need to spend hours debugging data consistency issues with NoSQL systems. With easy scaling, there is no need to worry about the database being the bottleneck of your business growth. You can also reduce the time to debug anomalies with the SQL interface.

Consolidating the tech stack from a polyglot of data systems to TiDB means that engineers do not need multiple systems and unnecessary Enhance, Transform, and Load (ETL) pipelines while keeping the functionality and high availability needed in today's business environment.

Since there is strong consistency between the row store and column store, you ensure that users are making business decisions with the same data powering the application. By implementing a smart selection based on query pattern, the system automatically chooses to read from the row or column store to process query requests quickly and efficiently.

TiDB is readily available on multiple clouds, on VMs, and on Kubernetes. It can be run on-premises, in a self-managed cloud environment, or through our fully managed cloud option, TiDB Cloud.

# Customer Examples

## Pinterest

In 2012, sharded MySQL was introduced into the Pinterest tech stack where it was used to store the core data models including pins, boards, and users. Next, in 2013, HBase was introduced as the first NoSQL database system at Pinterest. Since then, HBase has been very popular and it has been serving many critical services. Based on HBase, they have built a columnar data store, graph service, and several other applications. In 2015, RocksDB was introduced as a high-performance C++ based storage engine, where it powers the Key-Value store, machine learning serving applications, and more. 2017 saw an increased number of customer requirements where they were asking more for a NoSQL database. Now, in 2021, they are again seeing many use cases that require a sweet spot between SQL and NoSQL. Essentially users want the scalability of NoSQL scalability, with SQL-like features.

TiDB was selected because it best met their requirements for stability and performance. Pinterest sees greatly reduced system complexity, and stronger consistency because TiDB provides distributed transactions. They also see reduced infrastructure costs, because of the reduced number of components.

## Streak

Streak automatically takes data from Gmail and creates a CRM pipeline. It allows users to see the pipeline by salesperson and by opportunity size. Since 2011, they have grown to 6000+ customers worldwide. They have a small team, about 5 back-end engineers and 5 front-end engineers. Consolidating database systems and having fewer pieces that people need to worry about improves developer velocity.

They would like to answer questions such as "How did things move between stages in a pipeline?" or "To whom are items assigned in a pipeline?". For example, in the case of deal management, Streak provides functionality to allow users to find leads that are connected to a particular salesperson. They also allow the users to order the leads by customized field(s). Within each pipeline, the number of leads can be in the hundreds or thousands. It is difficult to efficiently index all the fields. TiDB's distributed parallel SQL execution allows them to return the result within the desired latency.

Streak also stores an index on email metadata that is used to do email sharing within teams, so if you add a particular email thread to the sales lead, they show that in a timeline view. That is

now moved over to TiDB to make it easier to work with in terms of being able to use a SQL prompt to debug when messages are missing.

## Additional Resources

[Simplifying Database Management in a Highly Automated and Scalable Cloud Environment with Google Cloud](#)

[SQL Plan Management: Never Worry About Slow Queries Again](#)