# NoSQL Case Study: Google Cloud Datastore

Introduction

Google Cloud Datastore is a NoSQL document database built for automatic scaling, high performance, and ease of application development. Cloud Datastore features include:

- **Atomic transactions**
  Cloud Datastore can execute a set of operations where either all succeed, or none occur.
- **High availability of reads and writes**
  Cloud Datastore runs in Google data centers, which use redundancy to minimize impact from points of failure.
- **Massive scalability with high performance**
  Cloud Datastore uses a distributed architecture to automatically manage scaling. Cloud Datastore uses a mix of indexes and query constraints so your queries scale with the size of your result set, not the size of your data set.
- **Flexible storage and querying of data**
  Cloud Datastore maps naturally to object-oriented and scripting languages, and is exposed to applications through multiple clients. It also provides a SQL-like query language.
- **Balance of strong and eventual consistency**
  Cloud Datastore ensures that entity lookups by key and ancestor queries always receive strongly consistent data. All other queries are eventually consistent. The consistency models allow your application to deliver a great user experience while handling large amounts of data and users.
- **Encryption at rest**
  Cloud Datastore automatically encrypts all data before it is written to disk and automatically decrypts the data when read by an authorized user.
- **Fully managed with no planned downtime**
  Google handles the administration of the Cloud Datastore service so you can focus on your application. Your application can still use Cloud Datastore when the service receives a planned upgrade.

Architecture
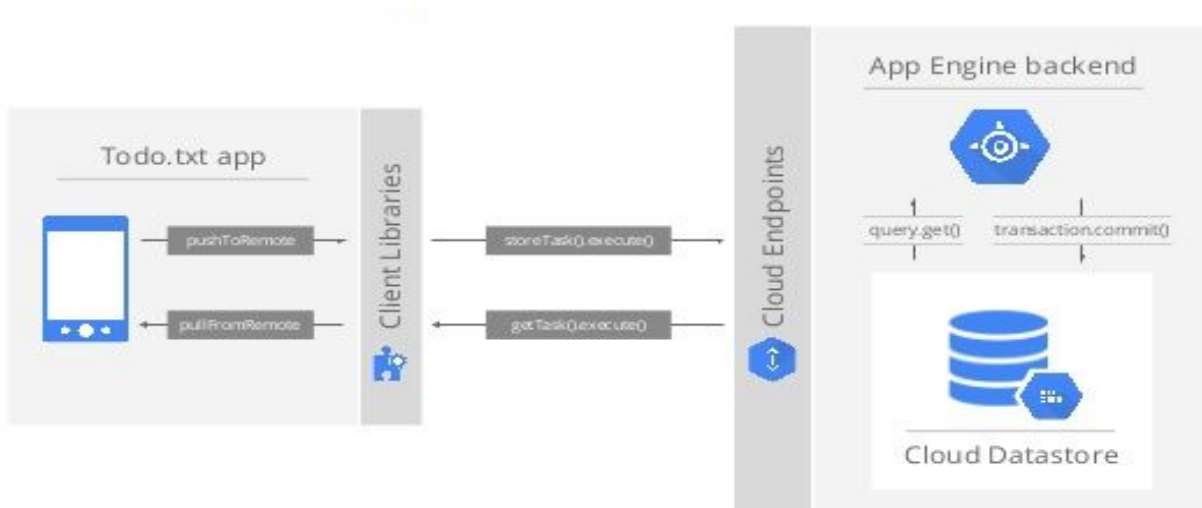
NoSQL Model

While the Cloud Datastore interface has many of the same features as traditional databases, as a NoSQL database it differs from them in the way it describes relationships between data objects. nlike rows in a relational database table, Cloud Datastore entities of the same kind can have different properties, and different entities can have properties with the same name but different value types. These unique characteristics imply a different way of designing and managing data to take advantage of the ability to scale automatically. In particular, Cloud Datastore differs from a traditional relational database in the following important ways:

- Cloud Datastore is designed to automatically scale to very large data sets, allowing applications to maintain high performance as they receive more traffic:
- Cloud Datastore writes scale by automatically distributing data as necessary.
- Cloud Datastore reads scale because the only queries supported are those whose performance scales with the size of the result set (as opposed to the data set). This means that a query whose result set contains 100 entities performs the same whether it searches over a hundred entities or a million. This property is the key reason some types of queries are not supported.
- Because all queries are served by previously built indexes, the types of queries that can be executed are more restrictive than those allowed on a relational database with SQL. In particular, Cloud Datastore does not include support for

join operations, inequality filtering on multiple properties, or filtering on data based on results of a subquery.
● Unlike traditional relational databases which enforce a schema, Cloud Datastore doesn't require entities of the same kind to have a consistent property set (although you can choose to enforce such a requirement in your own application code).

Advantages

● Product catalogs that provide real-time inventory and product details for a retailer.
● User profiles that deliver a customized experience based on the user's past activities and preferences.
● Transactions based on ACID properties, for example, transferring funds from one bank account to another.

Disadvantages

● Cloud Datastore is not ideal for every use case.
● Cloud Datastore is not a relational database
● Cloud Datastore is not an effective storage solution for analytic data.
Comparison between Datastore & Big Table

Comparison between Datastore and BigTable

Datastore	BigTable
Extremely high availability with replication and data synchronization	Doesn't replicate data across zones or regions, making it faster and more efficient, and costs are much lower
More expensive	Less expensive
Slower writing data due to synchronous replication	Faster writing data due to no replication

Comparison with Google Cloud Databases

Object/File	Relational	NoSQL		Cache
Cloud Storage	Cloud SQL	Cloud Datastore	Cloud Bigtable (Beta)	App Engine Memcache
Good for: Structured and unstructured binary or object data	**Good for:** Web frameworks, existing applications	**Good for:** Mobile/Web, App Engine	**Good for:** Heavy read + write, events	**Good for:** Web/mobile applications, gaming
Such as: Images, large media files, backups	**Such as:** User credentials, customer orders	**Such as:** User profiles, catalogs	**Such as:** AdTech, Financial, IoT	**Such as:** Game state, user sessions

Companies using Cloud Datastore

- Programming Hub: Cloud based programming
- Foodit: Creates Mobile App and Websites for Restaurants
- Skotty: E2E File Encryption
- Shade: UV sensor