

History of Manhattan

We'll start with Manhattan, Twitter's in-house distributed key-value store. Manhattan is responsible for storing tweets, accounts, direct messages, and more. They developed it after having scalability issues with Cassandra. Cassandra is an open source NoSQL database with many servers that handle large volumes of data.

Since the very nature of Twitter is built around the idea of tweets being served in real-time, a low-latency database is paramount to its success.

Twitter built Manhattan specifically to address requirements like reliability, availability, low-latency, and scalability, but also extensibility, operability, and developer productivity. By providing a large-scale database that is built for the future, Twitter can ensure that developers will be able to store and retrieve whatever data they need without having to worry about it.

Manhattan is separated into four main layers. From top to bottom they are:

- * **Interfaces**: How a customer interacts with the system.
- * **Storage services**: Additional, developer-centric features like: Batch Hadoop importing, Strong Consistency service, Timeseries Counters service.
- * **Core**: The most critical aspect of the system, it allows for adaptability as well as handling all sorts of conflicts.
- * **Storage engines**: One of the lowest levels of storage, this informs how data is kept on a disc and the data structures used to store information in memory.

