I decided to pick X messenger (formerly known as Twitter) for my research. Let’s go through the key points.

Architecture:

Initially, X was built as a monolithic. However, as the platform grew, X transitioned to a microservices architecture to improve scalability and maintainability.

Code languages:

For the frontend X uses JS and React and for the backend X uses several languages such as C++, Scala, Ruby and Java.

Databases:

Uses 8 different databases for different purposes, let’s go over them shortly:

Hadoop - for social graph analysis, recommendations, trends, API analytics, user engagement prediction, ad targeting, ad analytics, tweet impressions processing, taking MySQL backups, Manhattan backups & storing front-end scribe logs.

MySQL & Manhattan - the primary data stores for storing user data.

Memcache, Redis - for caching

FlockDB - for storing social graph

MetricsDB - for storing platform data metrics

Blobstore - for storing images, videos & large binary objects.

Performance optimization:

Caching: as previously mentioned, Memcache and Redis are used to reduce the load on its databases and improve response times.

Asynchronous Processing: X uses asynchronous processing for tasks that can be deferred. By offloading tasks to background jobs, the main application can remain responsive to user requests.

Used materials:  
[X’s Wikipedia page](https://en.wikipedia.org/wiki/Twitter)

[What Database Does Twitter Use? - A Deep Dive - Scaleyourapp](https://scaleyourapp.com/what-database-does-twitter-use-a-deep-dive/)