



JavaScript Runtime

On Server Side

# Database mongoose vs Knex.js

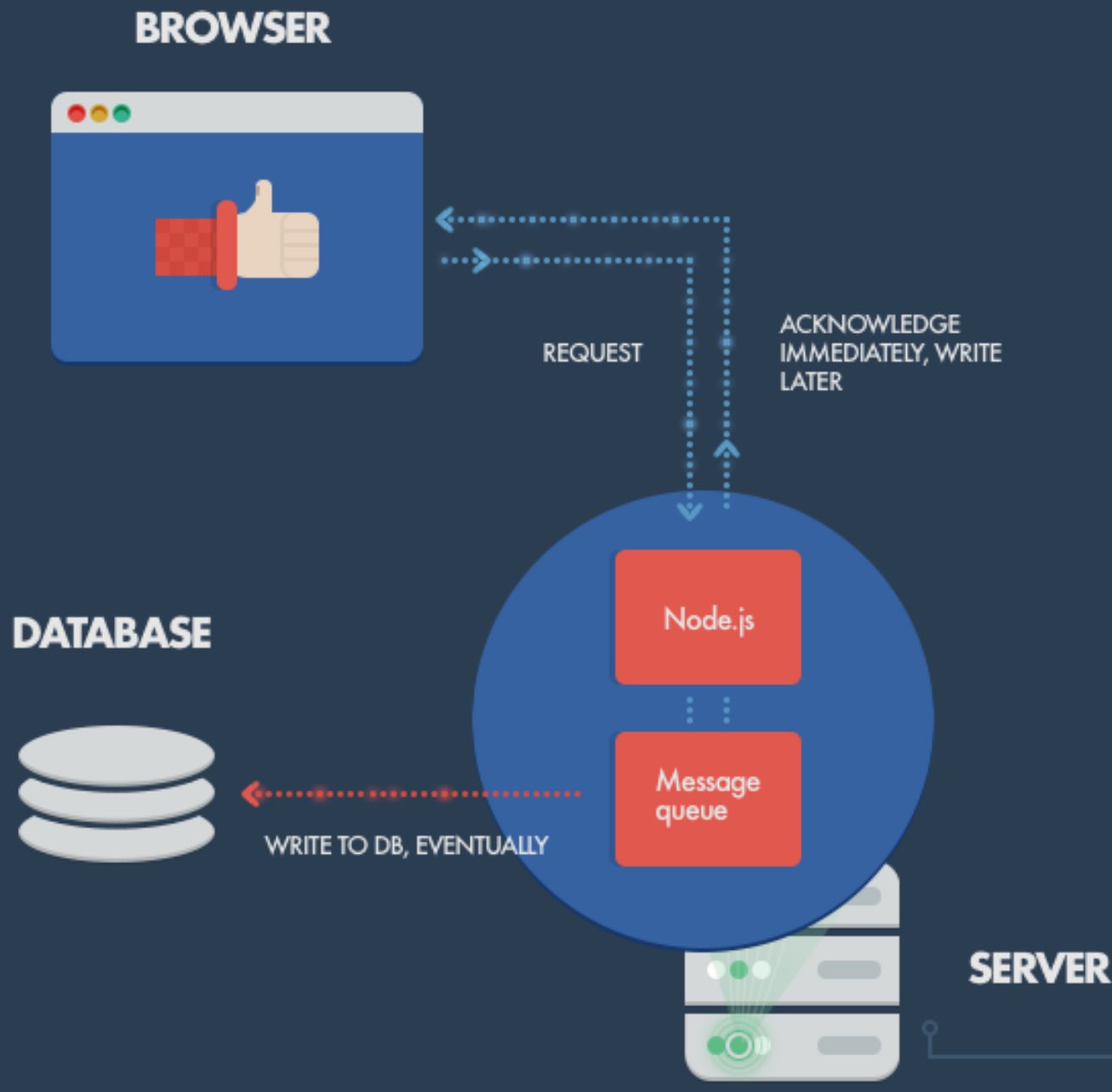


# Database integration

- Cassandra
- Couchbase
- CouchDB
- LevelDB
- MySQL
- MongoDB
- Oracle
- PostgreSQL
- Redis
- SQL Server
- SQLite
- ElasticSearch



ture



# Mongodb

- Module: `mongoose`
- Installation: `npm install mongoose @types/mongoose`



# Example - Mongodb

- <https://www.npmjs.com/package/mongoose>
- <https://mongoosejs.com/docs/guide.html>



# MySQL

- Module: `mysql`
- Installation: `npm install mysql`



# Example - MySQL

```
var mysql = require('mysql')
var connection = mysql.createConnection({
  host      : 'localhost',
  user      : 'dbuser',
  password  : 's3kre337',
  database  : 'my_db'
});

connection.connect()

connection.query('SELECT 1 + 1 AS solution', function (err, rows, fields) {
  if (err) throw err

  console.log('The solution is: ', rows[0].solution)
})

connection.end()
```





# Knex.js - A SQL Query Builder for Javascript

- Knex.js is a "batteries included" SQL query builder for Postgres, MSSQL, MySQL, MariaDB, SQLite3, Oracle, and Amazon Redshift designed to be flexible, portable, and fun to use



Nodemy



KNEX.JS



# Knex.js - A SQL Query Builder for Javascript

- Cleaner async flow control with callbacks as well as a promise style
- Full featured query and schema builders
- Transaction support (with savepoints)
- Connection pooling and standardized responses between different query clients and dialects



# References

- <https://expressjs.com/en/guide/database-integration.html>
- <https://knexjs.org/>
- <https://www.getpostman.com/>

