

SERVER ("BACK END")

CLIENT ("FRONT END")



Machine 1 (computer + operating system)

Machine 2 (computer + operating system)



Filesystem (serial data)

static files
(images, .html docs,
.js files, etc.)

```
101110001110101101
010010110100101001
101010101011010101
011010010101101010
101101001010110101
01011010
```

raw database data
(e.g. BSON, .sql)

```
101110001110101101
010010110100
```

FS
I/O

FS
I/O



DBMS:
mongod / postgres
process (C/C++)

translate query into binary
read/write and send back
result to requester

Db Actions

Db Actions

mongo process

```
> shell  
(in terminal)
```



Node.js process
(C/C++ runs V8, libev, libeio)

JavaScript (being executed by **V8**)

JS



Express.js (middleware framework)



morgan (logger middleware)

next();

express.static (middleware)

next();

body-parser (middleware)

next();

express.Router (middleware)

next(err);

error handling middleware

Mongoose (ODM) / **Sequelize** (ORM)



m

model

schema

instances

Db Actions

Db Actions

```
$ console/shell (in terminal)
```

HTTP

Chrome process
(C/C++ runs V8, DOM, etc.)



www.mycoolapp.com/somepath

Address bar (makes GET requests)

DOM (node tree, what gets rendered)

html

body

h1

button

```
> console (in dev tools)
```