

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.)

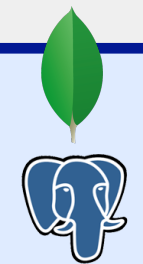
101110001110101101
010010110100101001
101010101011010101
0110100101011

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 Protocol

Db Protocol

mongo / psql process

> shell
(in terminal)

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

JavaScript (being executed by **V8**)

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)

schema

model

instances

mongodb / pg (database driver)

\$ 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

JavaScript (executed by **V8**)

Front-end tools (**jQuery**) or
frameworks (**Angular.js**)

factory

controller A

controller B

\$scope
(A)

\$scope
(B)

> console (in dev tools)