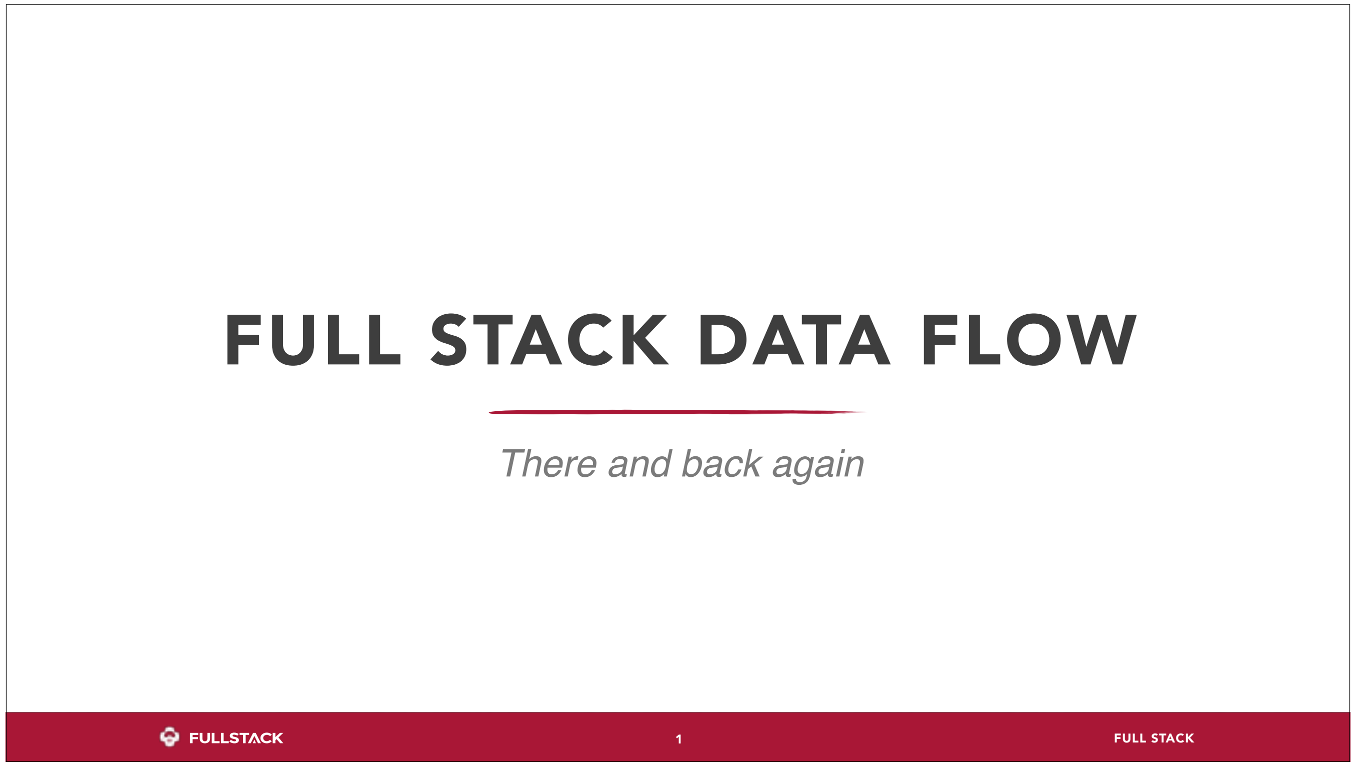
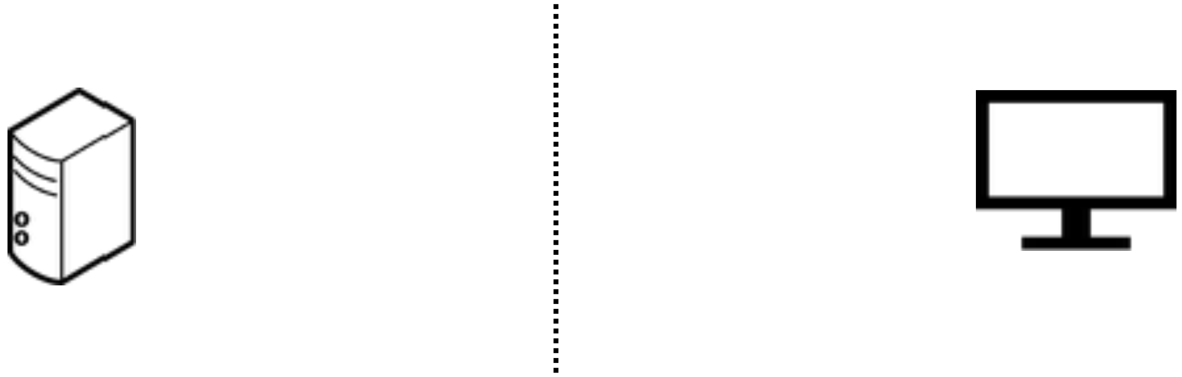
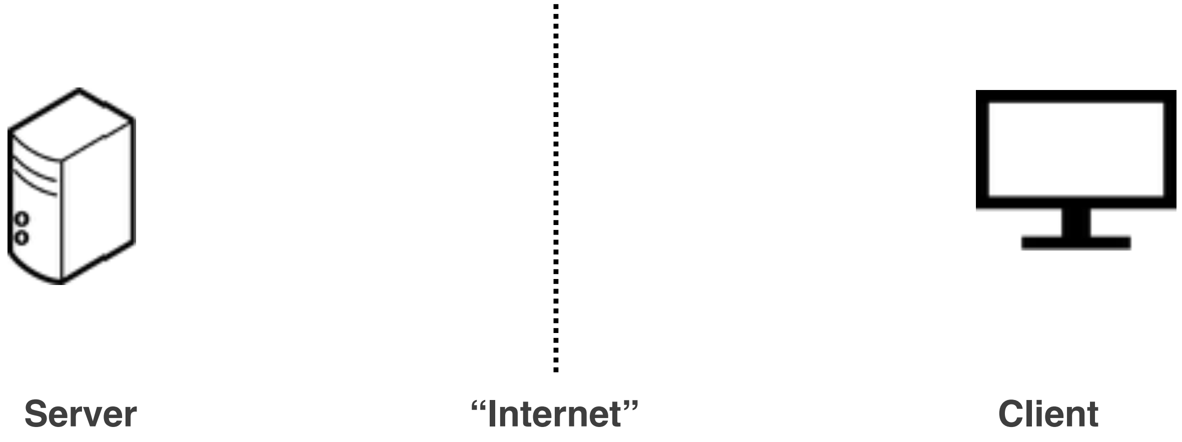


FULL STACK DATA FLOW

There and back again

two players - client and server

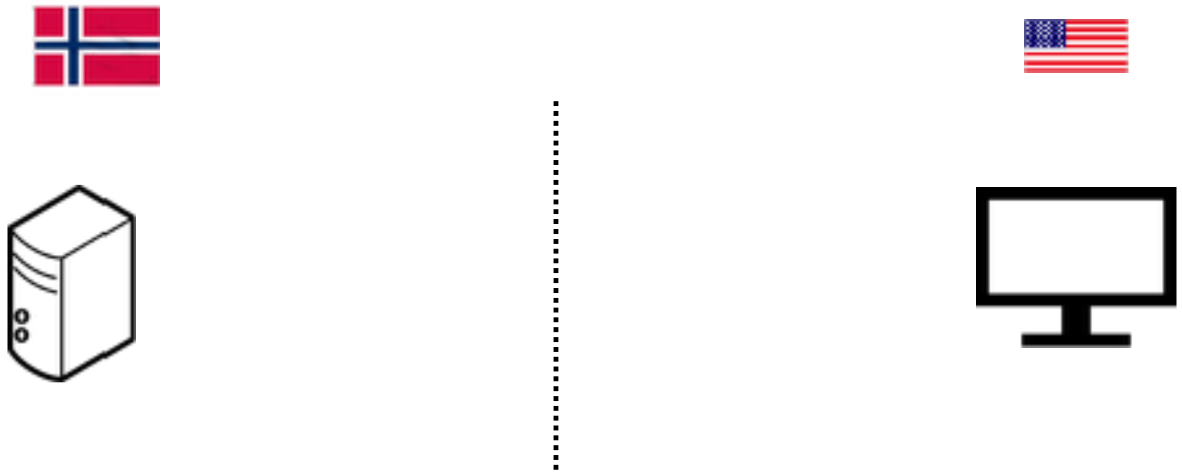




Server

“Internet”

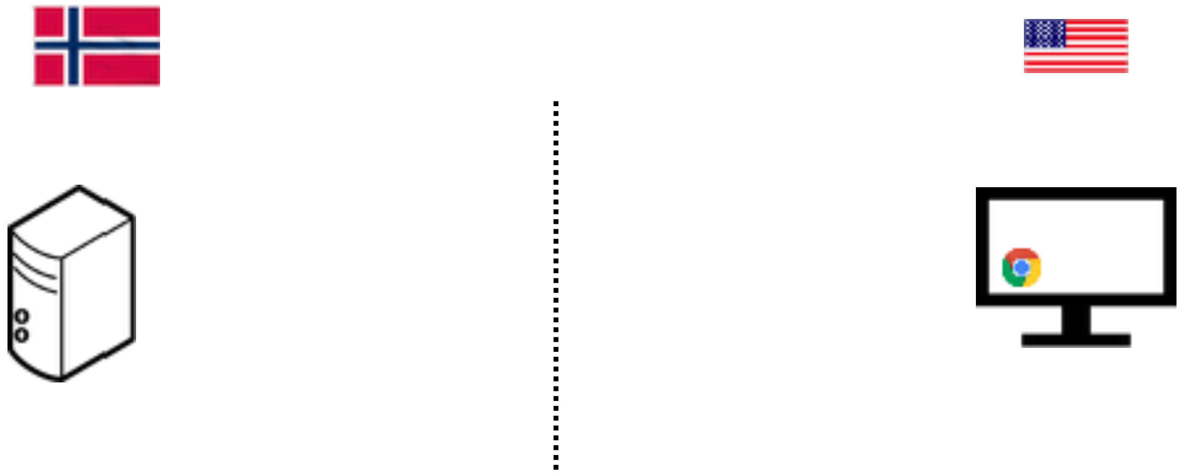
Client



Server

“Internet”

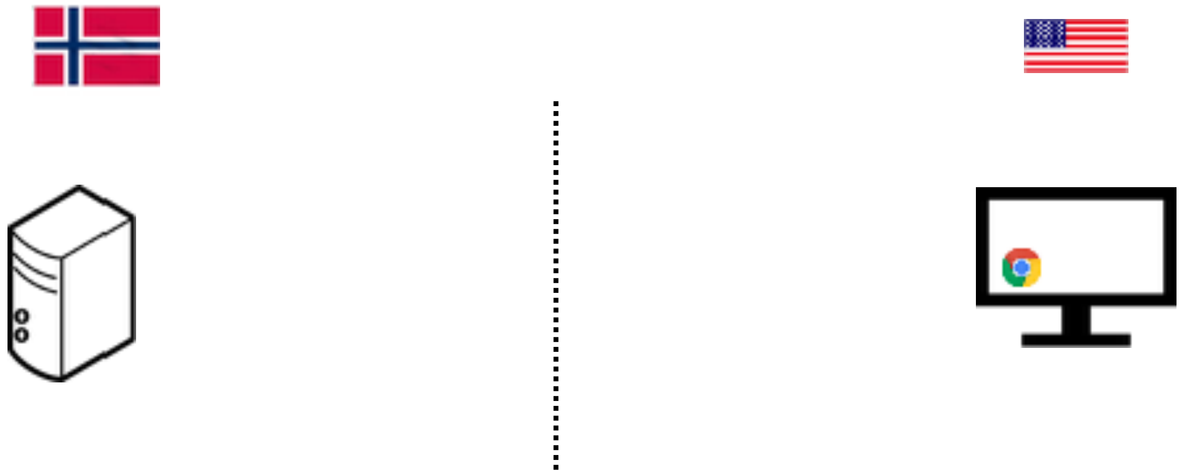
Client



Server

“Internet”

Client



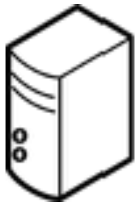
Server

“Internet”

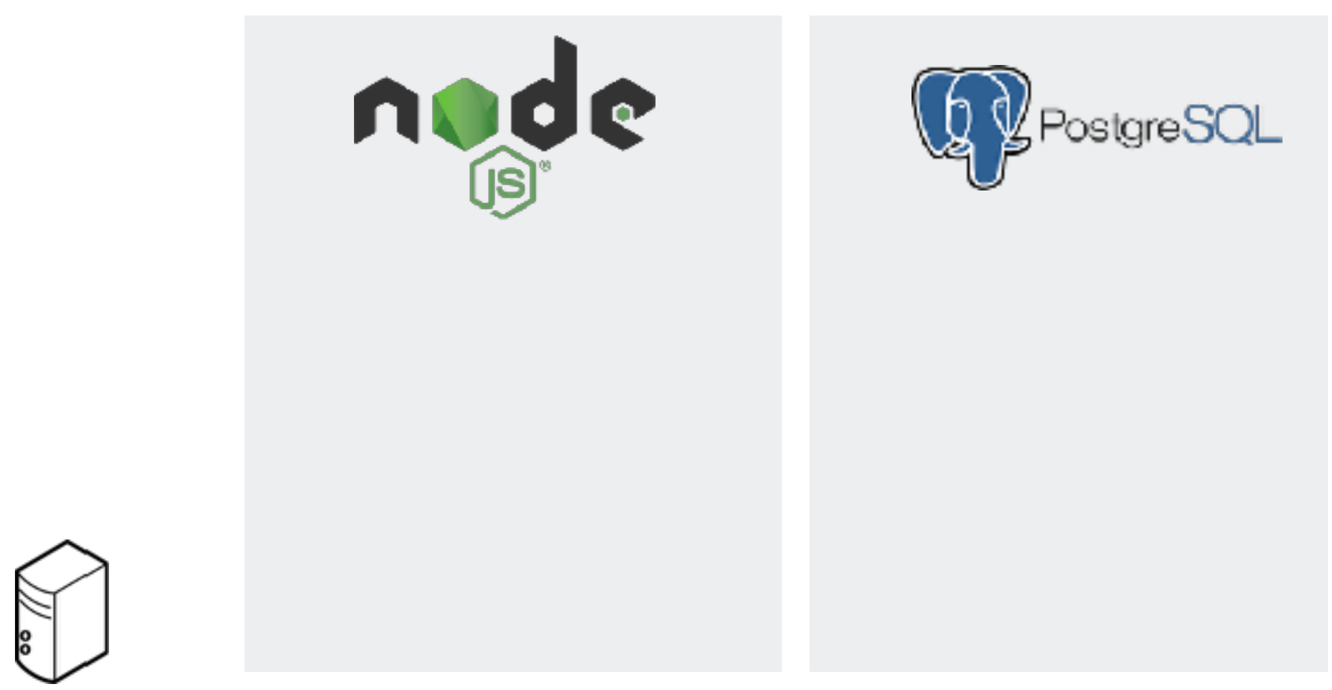
Client

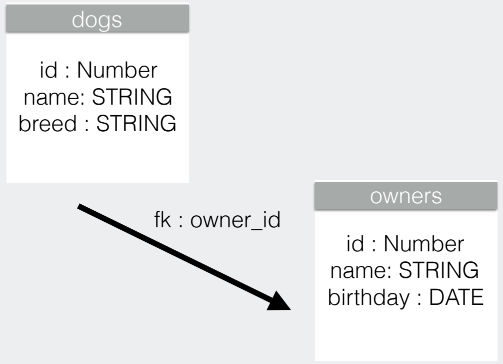
node
JS
PostgreSQL

- A JavaScript runtime built on Chrome’s V8 JS engine
- Allows execution of JS on an operating system



- **An open source, object-relational database system**


node
JS
PostgreSQL



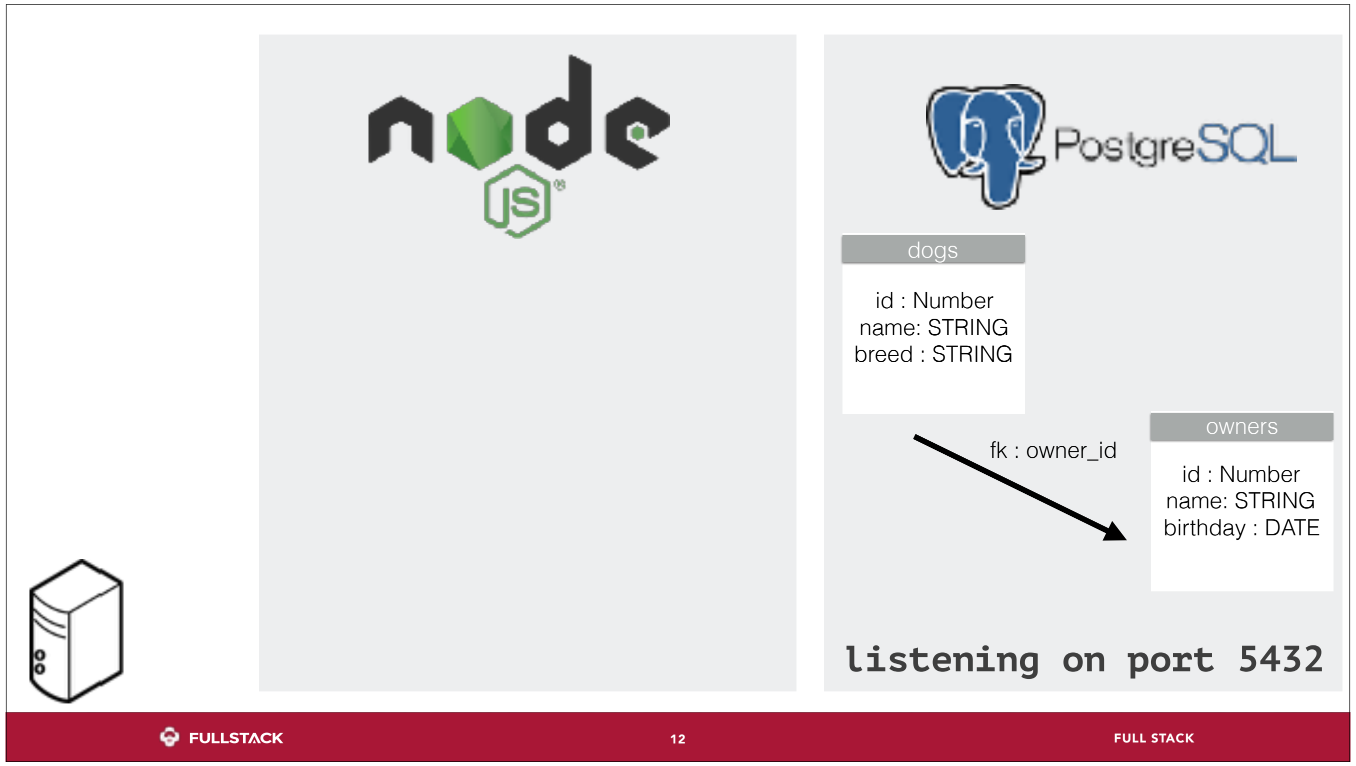
dogs

id : Number
name: STRING
breed : STRING

fk : owner_id

owners

id : Number
name: STRING
birthday : DATE



dogs

id : Number
name: STRING
breed : STRING

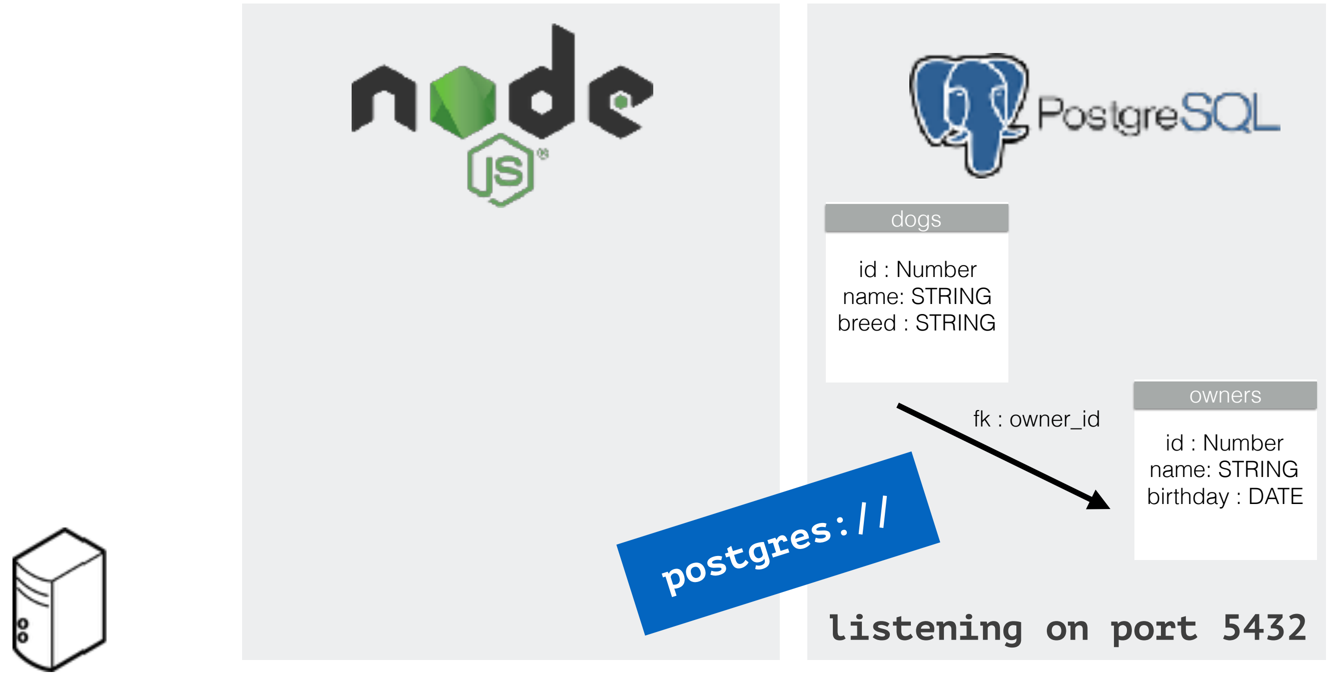
fk : owner_id

owners

id : Number
name: STRING
birthday : DATE

listening on port 5432

not exposed to the internet



dogs

id : Number
name: STRING
breed : STRING

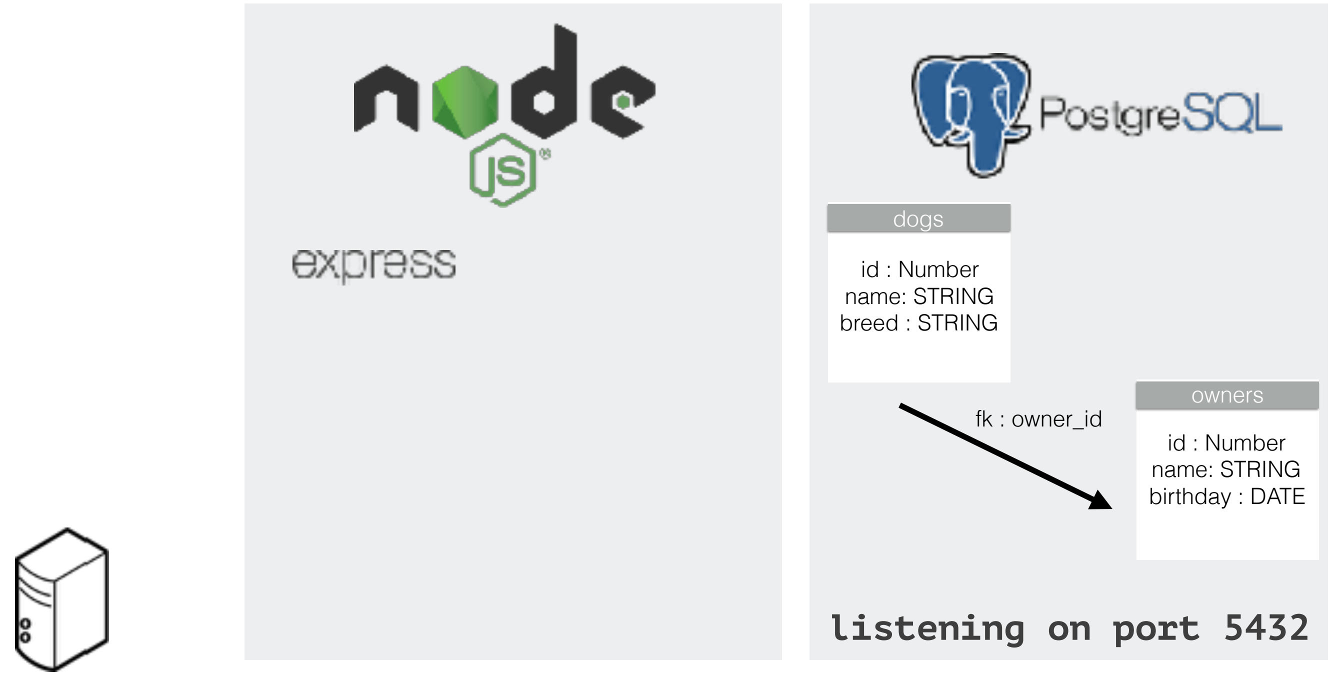
fk : owner_id

owners

id : Number
name: STRING
birthday : DATE

postgres://

`listening on port 5432`



dogs

id : Number
name: STRING
breed : STRING

fk : owner_id

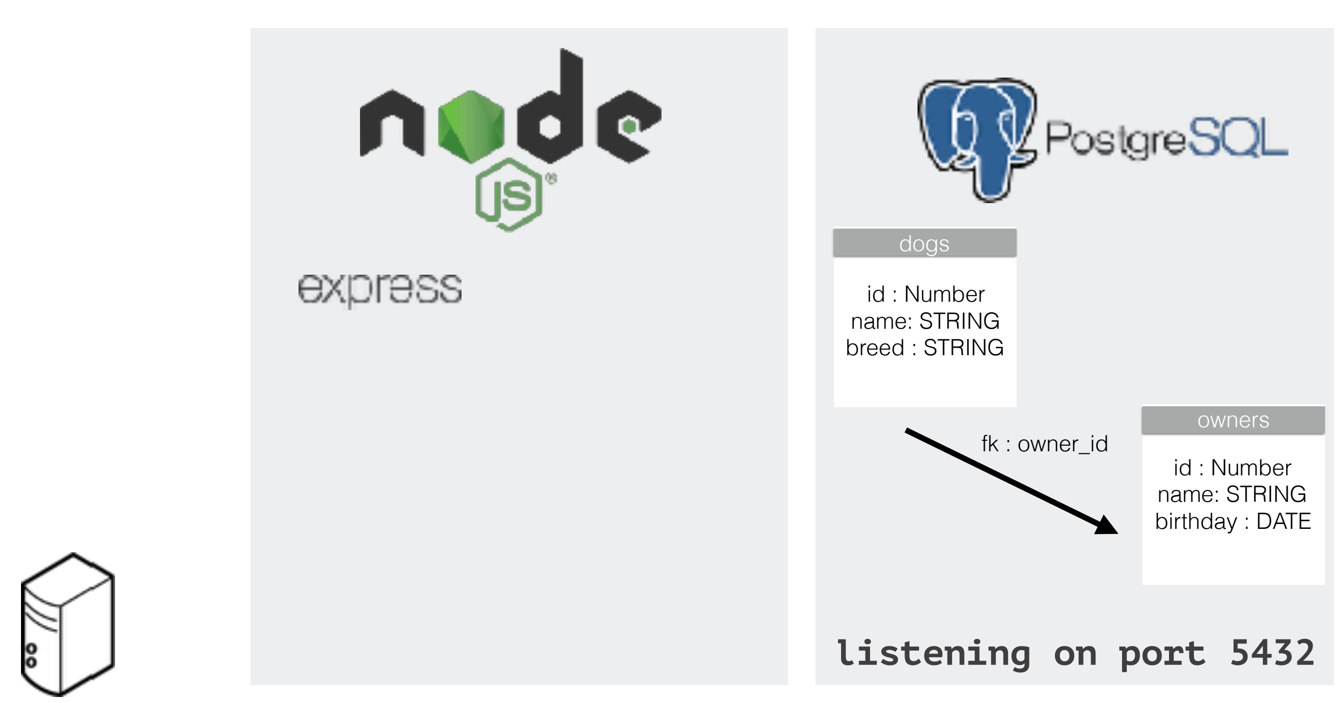
owners

id : Number
name: STRING
birthday : DATE

listening on port 5432

express

- A application written with Node.js
- Provides a framework for writing web servers



express

dogs

id : Number
name: STRING
breed : STRING

fk : owner_id

owners

id : Number
name: STRING
birthday : DATE

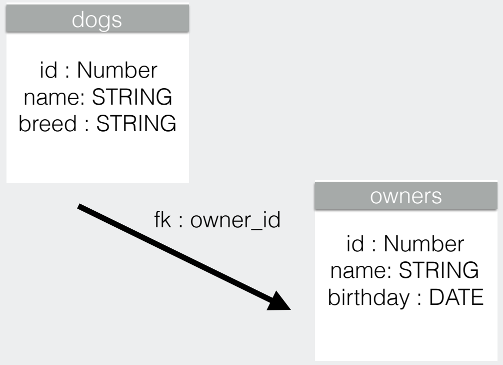
`listening on port 5432`

```
server/
  index.js
```

node js

express

PostgreSQL



dogs

id : Number
name: STRING
breed : STRING

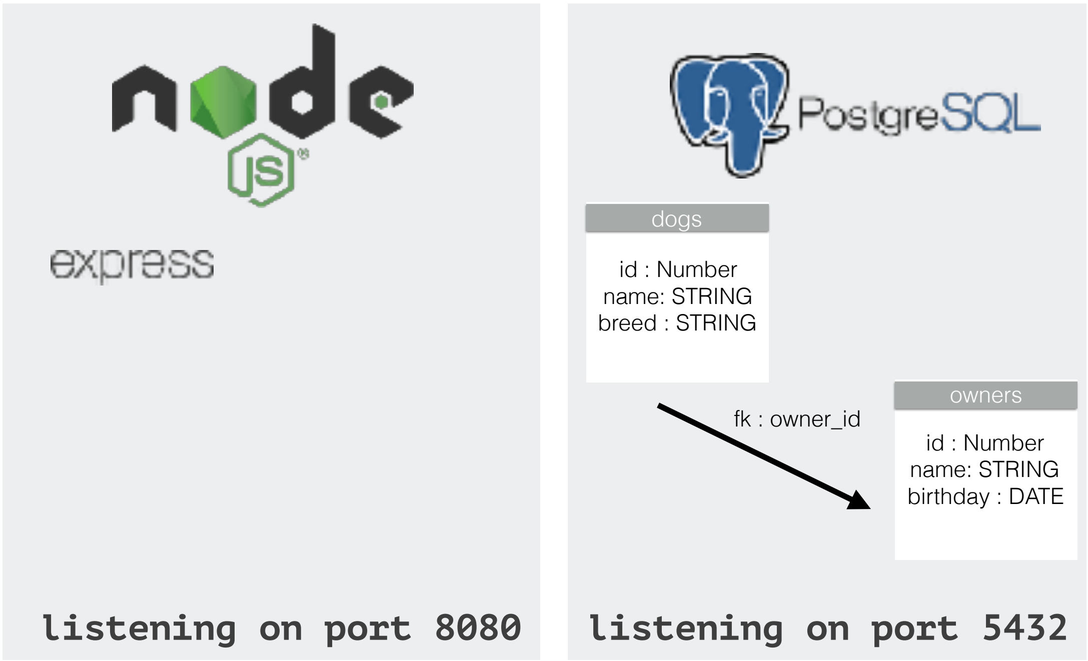
fk : owner_id

owners

id : Number
name: STRING
birthday : DATE

```
listening on port 5432
```

```
server/
  index.js
```



dogs
id : Number name: STRING breed : STRING

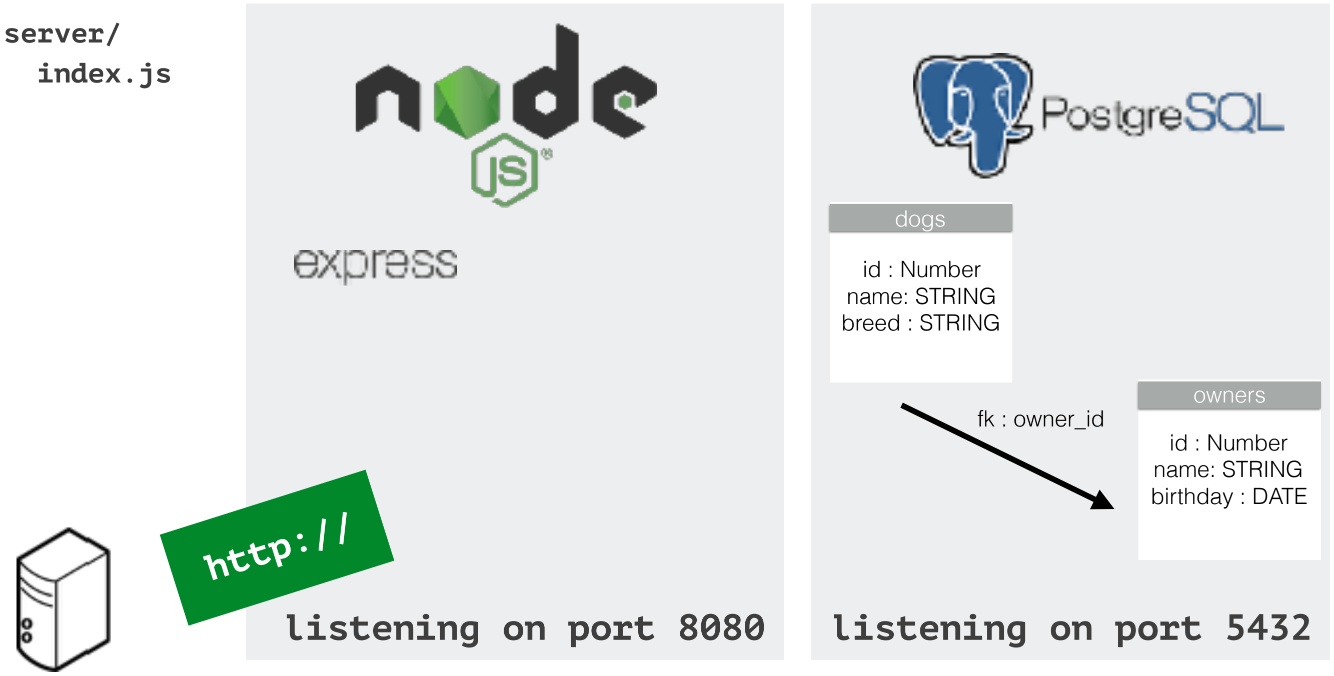
fk : owner_id

owners
id : Number name: STRING birthday : DATE

`listening on port 8080`

`listening on port 5432`

```
server/
  index.js
```



listening on port 8080

dogs

id : Number
name: STRING
breed : STRING

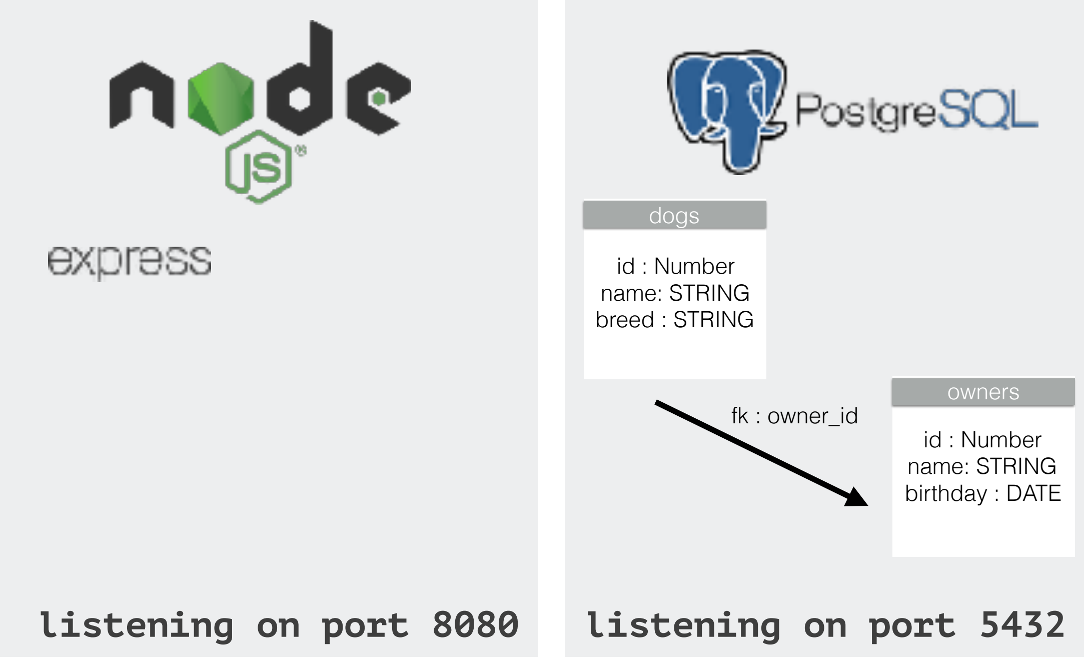
fk : owner_id

owners

id : Number
name: STRING
birthday : DATE

listening on port 5432

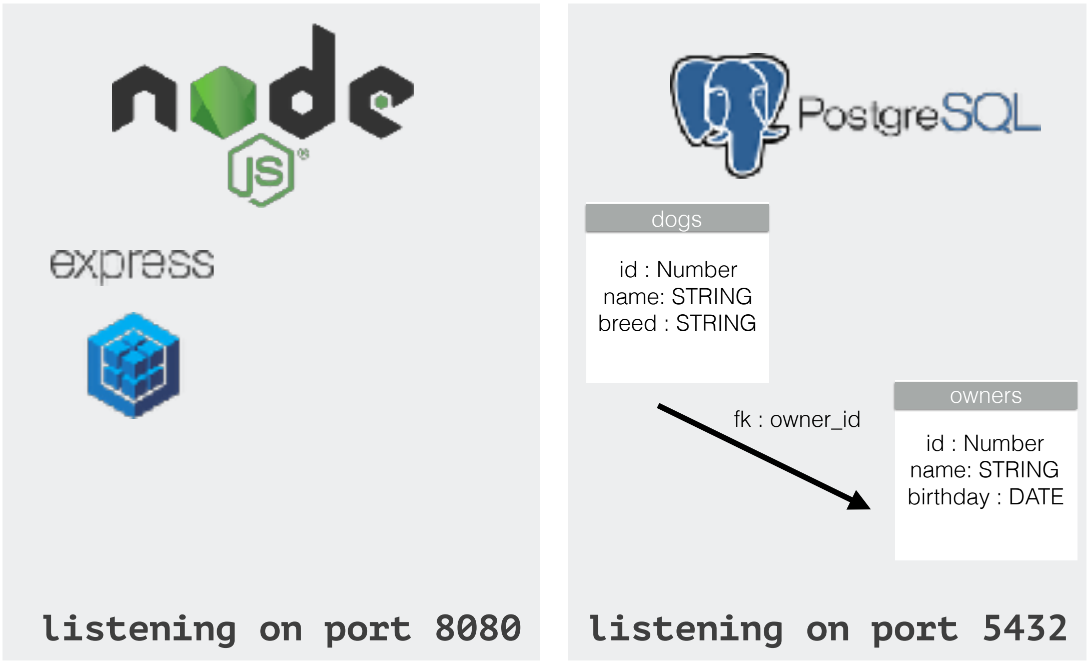
```
server/
  index.js
public/
  index.html
  style.css
```



listening on port 8080

listening on port 5432

```
server/
  index.js
public/
  index.html
  style.css
```



dogs

id : Number
name: STRING
breed : STRING

fk : owner_id

owners

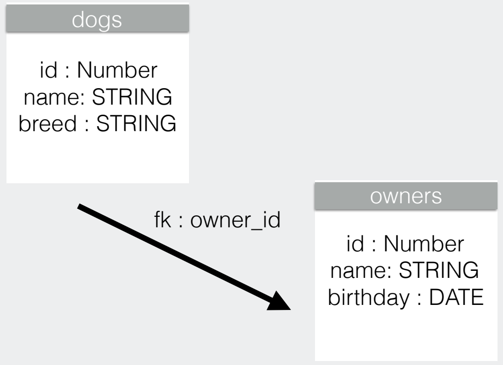
id : Number
name: STRING
birthday : DATE

listening on port 8080

listening on port 5432

- A application written with Node.js
- Provides a framework for interacting with a database from a Node application

```
server/
  index.js
public/
  index.html
  style.css
```



dogs

id : Number
name: STRING
breed : STRING

owners

id : Number
name: STRING
birthday : DATE

listening on port 8080

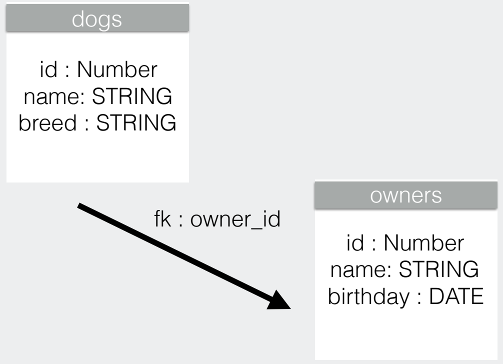
listening on port 5432

```
server/
  index.js
  db.js
public/
  index.html
  style.css
```

express

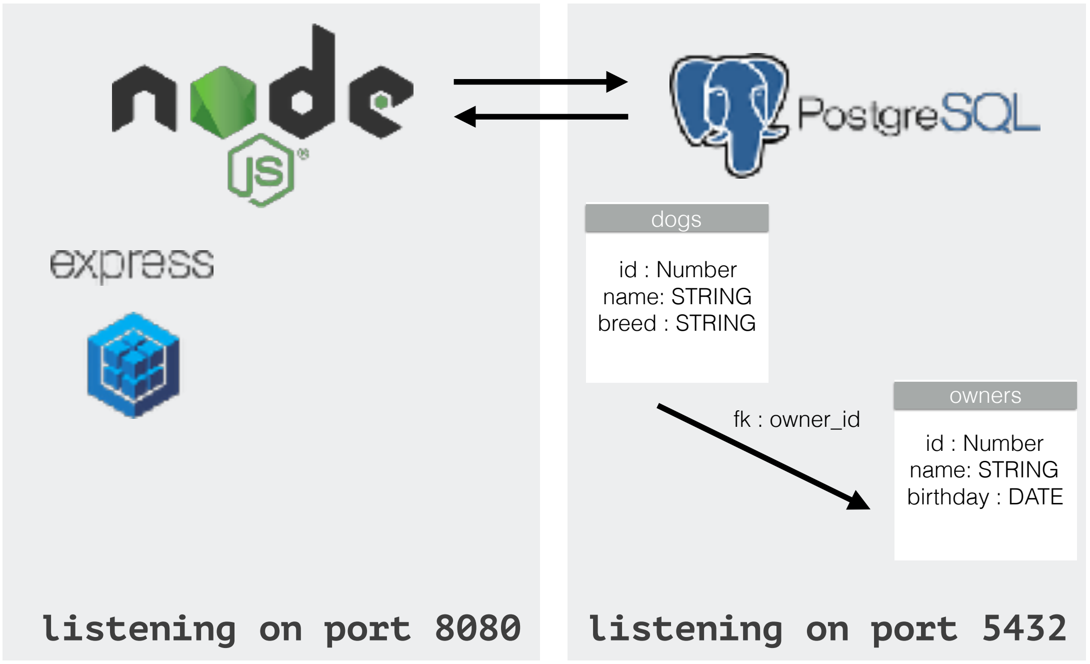
listening on port 8080

PostgreSQL



listening on port 5432

```
server/
  index.js
  db.js
public/
  index.html
  style.css
```

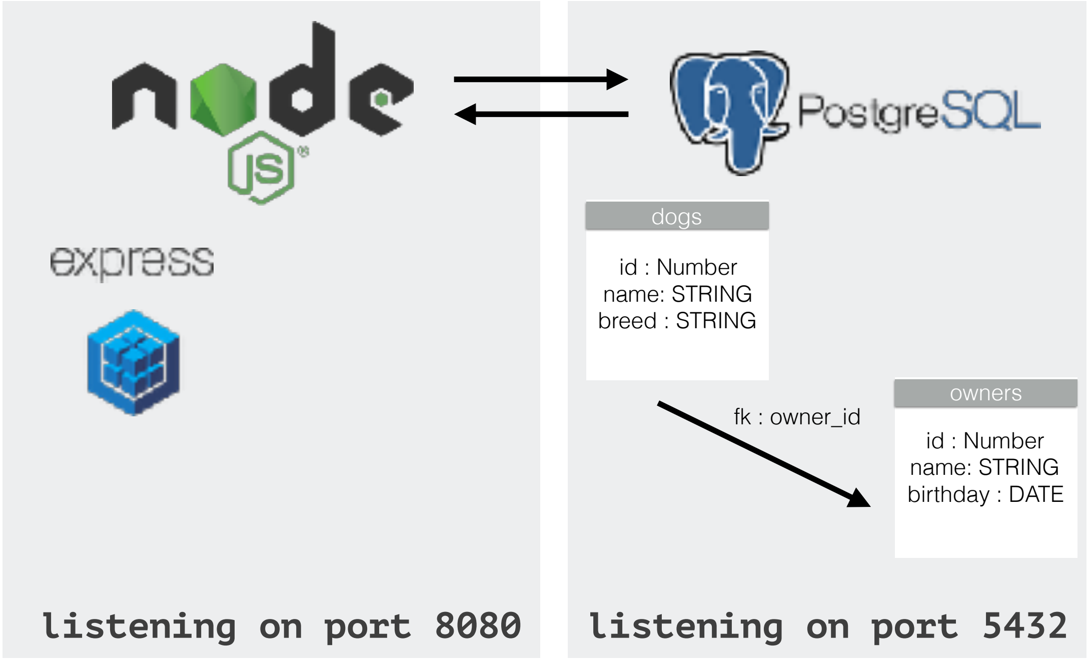


listening on port 8080

listening on port 5432

```
server/
  index.js
  db.js
public/
  index.html
  style.css

client/
  Footer.js
  Header.js
```



dogs

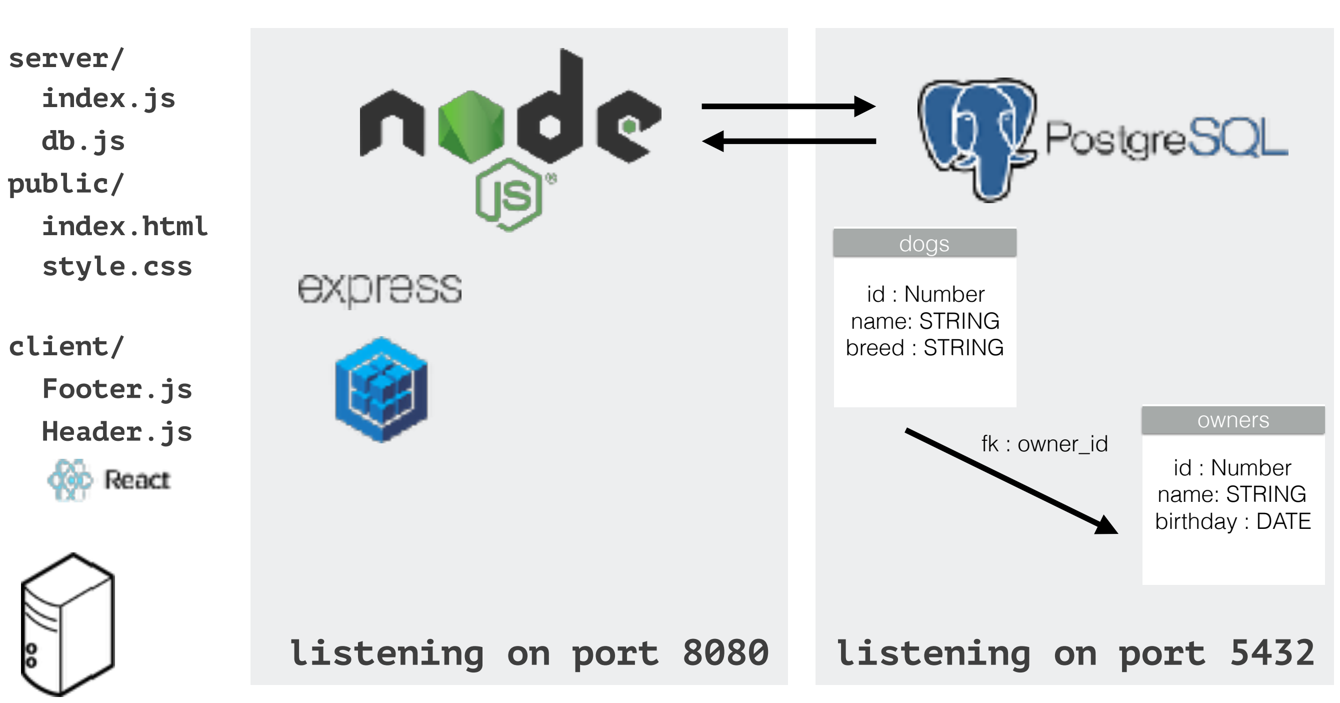
id : Number
name: STRING
breed : STRING

owners

id : Number
name: STRING
birthday : DATE

listening on port 8080

listening on port 5432



```
server/
  index.js
  db.js
public/
  index.html
  style.css

client/
  Footer.js
  Header.js
```

dogs

id : Number
name: STRING
breed : STRING

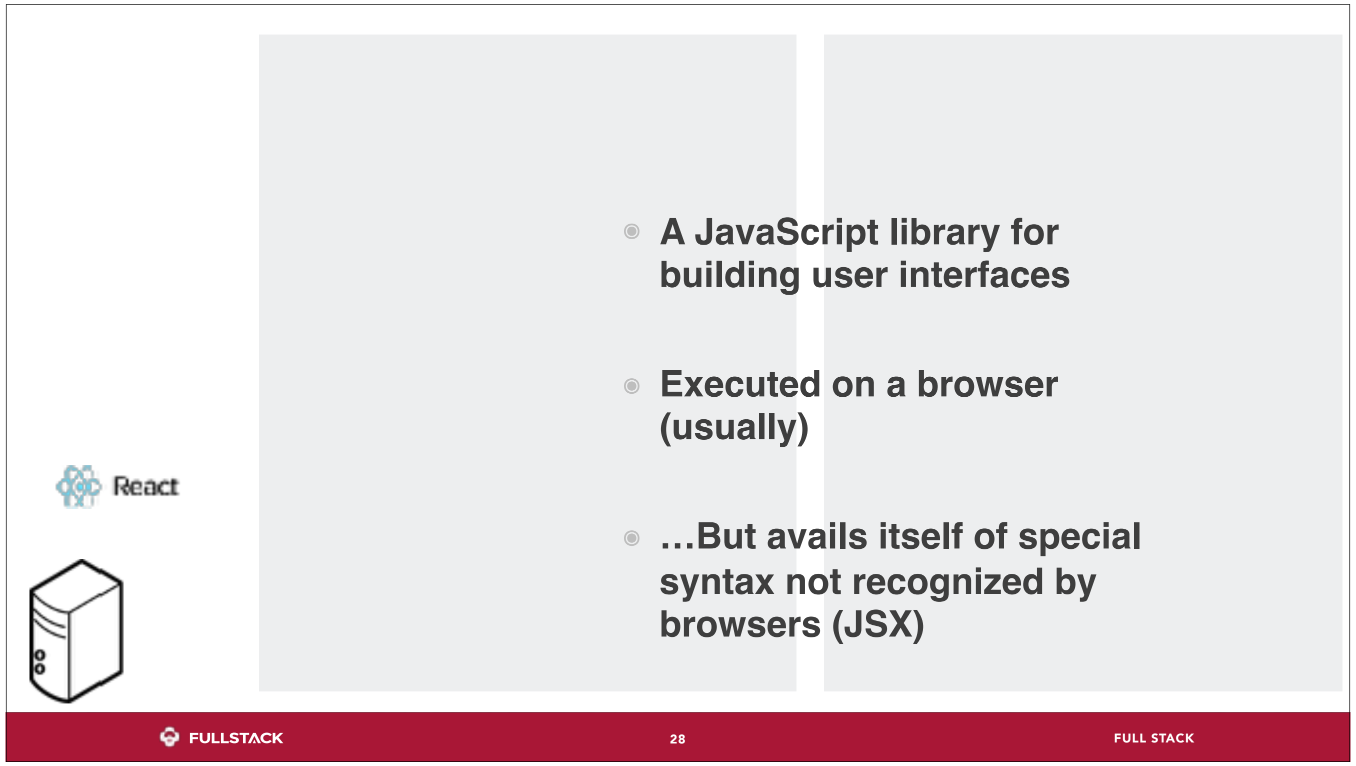
fk : owner_id

owners

id : Number
name: STRING
birthday : DATE

listening on port 8080

listening on port 5432



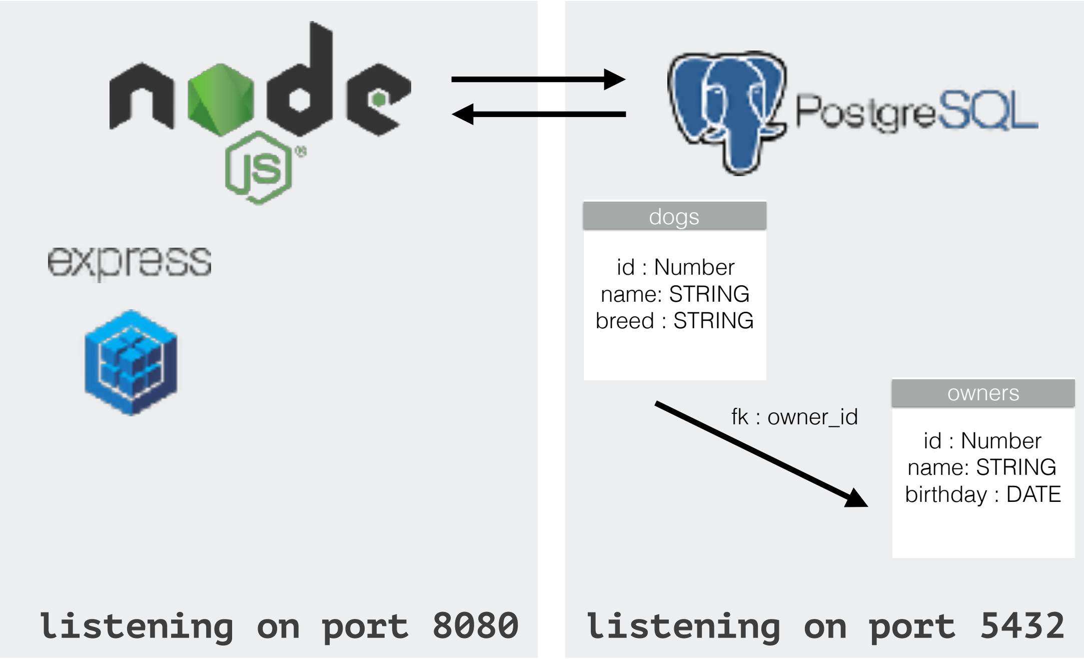
- A JavaScript library for building user interfaces
- Executed on a browser (usually)
- …But avails itself of special syntax not recognized by browsers (JSX)

*usually - React is actually flexible and can be used on the server, and in other environments as well, but for now we’re just interested in its use on the browser

```
server/
  index.js
  db.js
public/
  index.html
  style.css

client/
  Footer.js
  Header.js
```

React



dogs
id : Number name: STRING breed : STRING

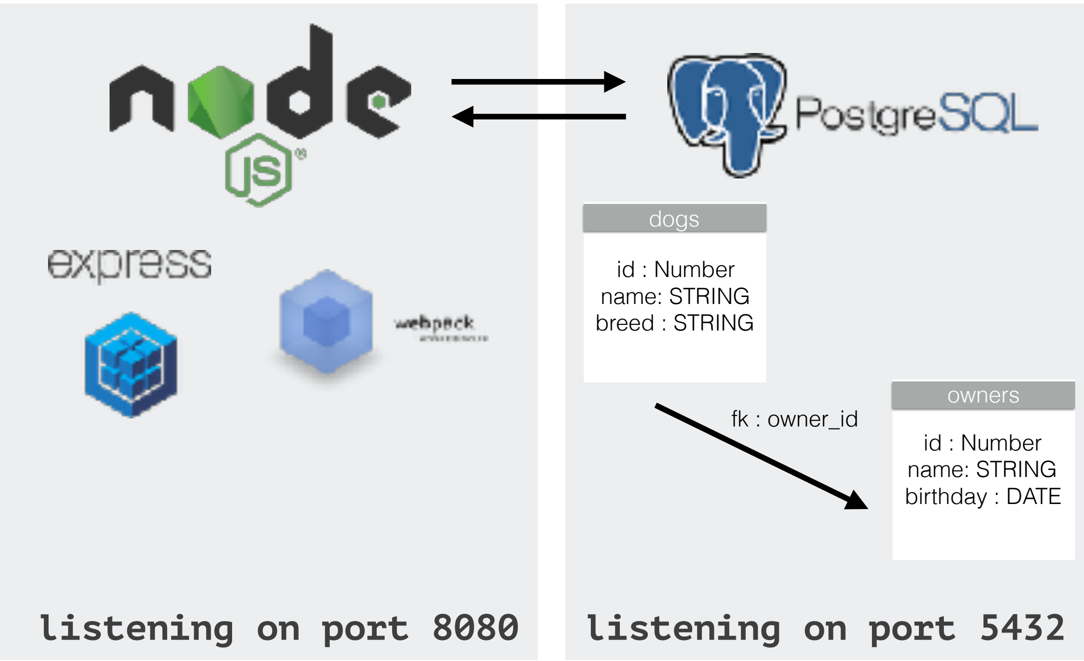
owners
id : Number name: STRING birthday : DATE

listening on port 8080

listening on port 5432

```
server/
  index.js
  db.js
public/
  index.html
  style.css

client/
  Footer.js
  Header.js
```



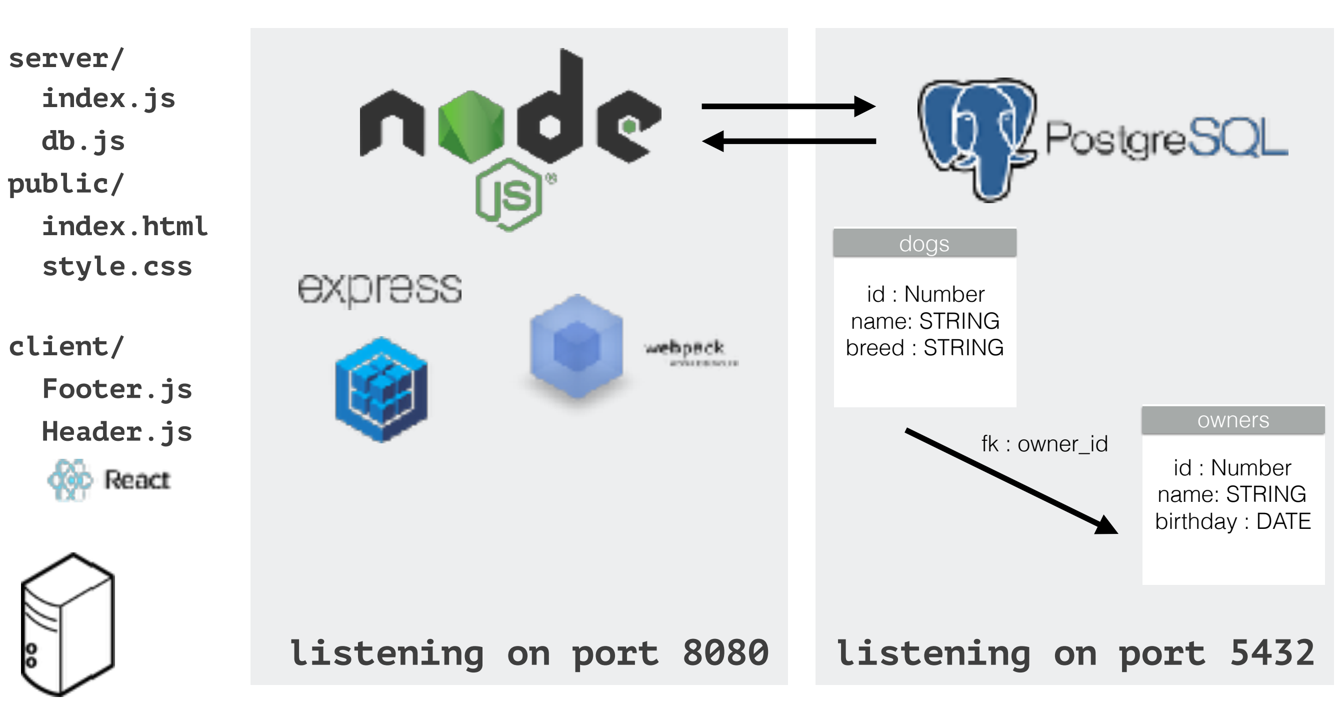
dogs
id : Number name: STRING breed : STRING

owners
id : Number name: STRING birthday : DATE

listening on port 8080

listening on port 5432

- A application written with Node.js
- Takes many .js files and “bundles” them into one
- Compiles special syntax like JSX into browser-safe JS



```
server/
  index.js
  db.js
public/
  index.html
  style.css

client/
  Footer.js
  Header.js
```

dogs

id : Number
name: STRING
breed : STRING

owners

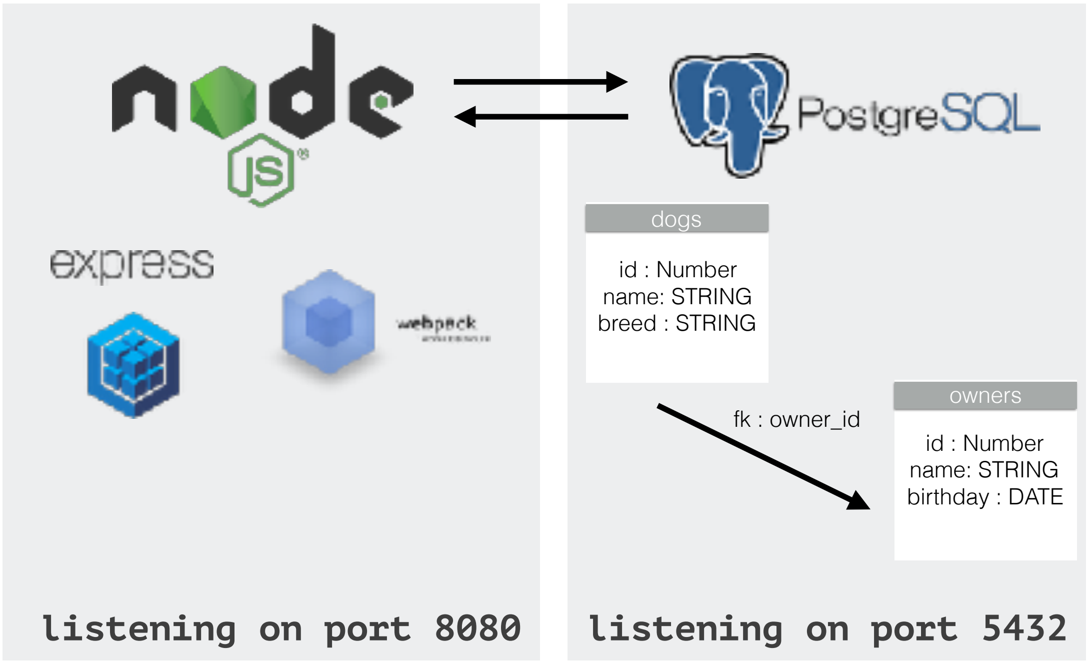
id : Number
name: STRING
birthday : DATE

listening on port 8080

listening on port 5432

```
server/
  index.js
  db.js
public/
  index.html
  style.css
  bundle.js
client/
  Footer.js
  Header.js
```

React



dogs

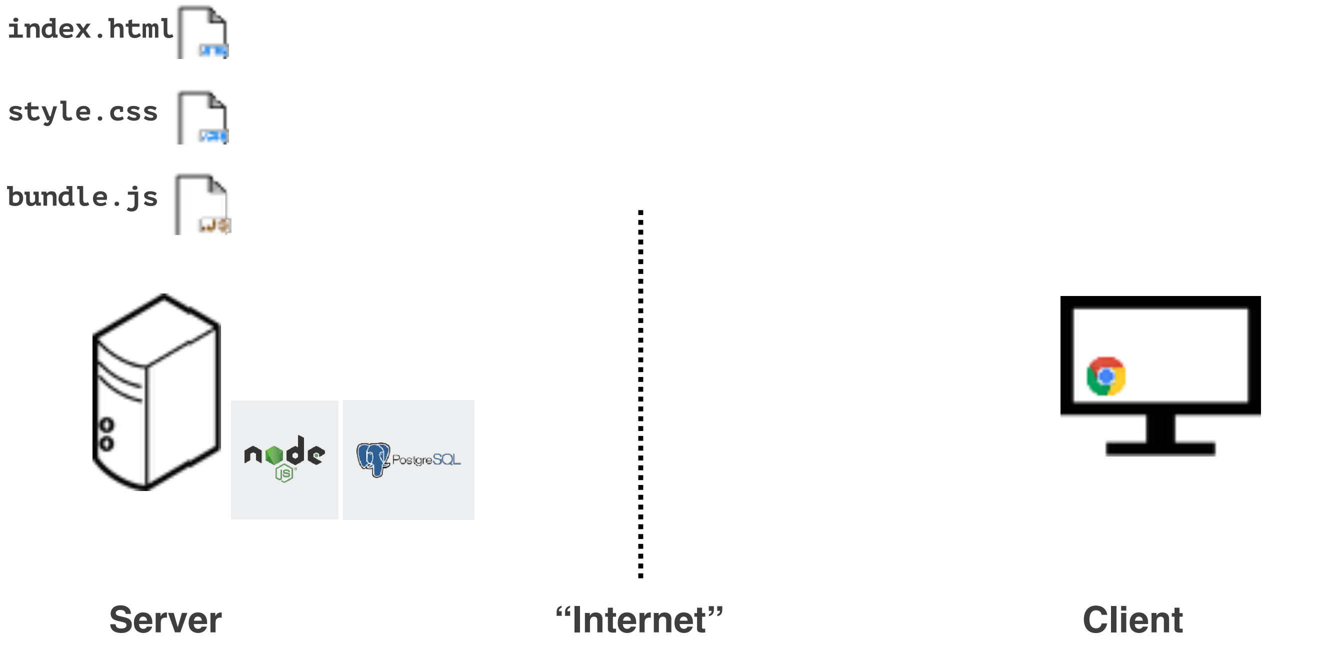
id : Number
name: STRING
breed : STRING

owners

id : Number
name: STRING
birthday : DATE

listening on port 8080

listening on port 5432



index.html

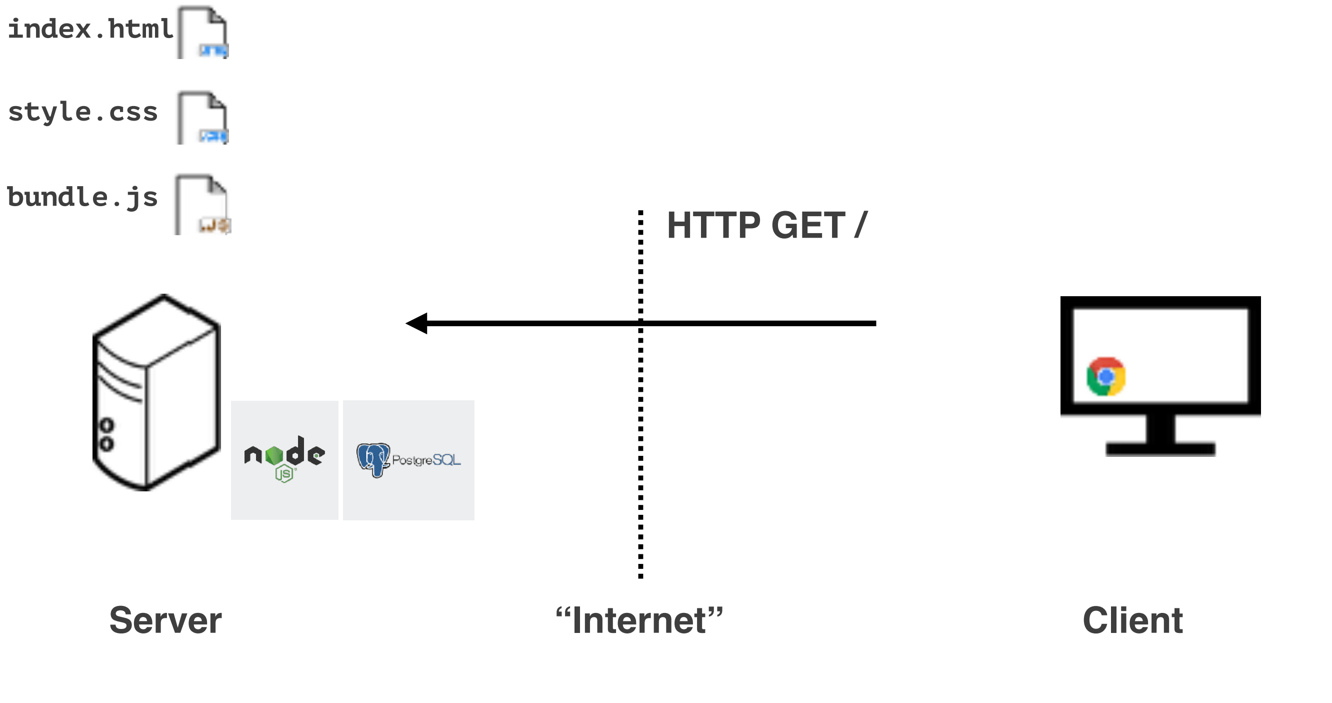
style.css

bundle.js

Server

“Internet”

Client



`index.html`

`style.css`

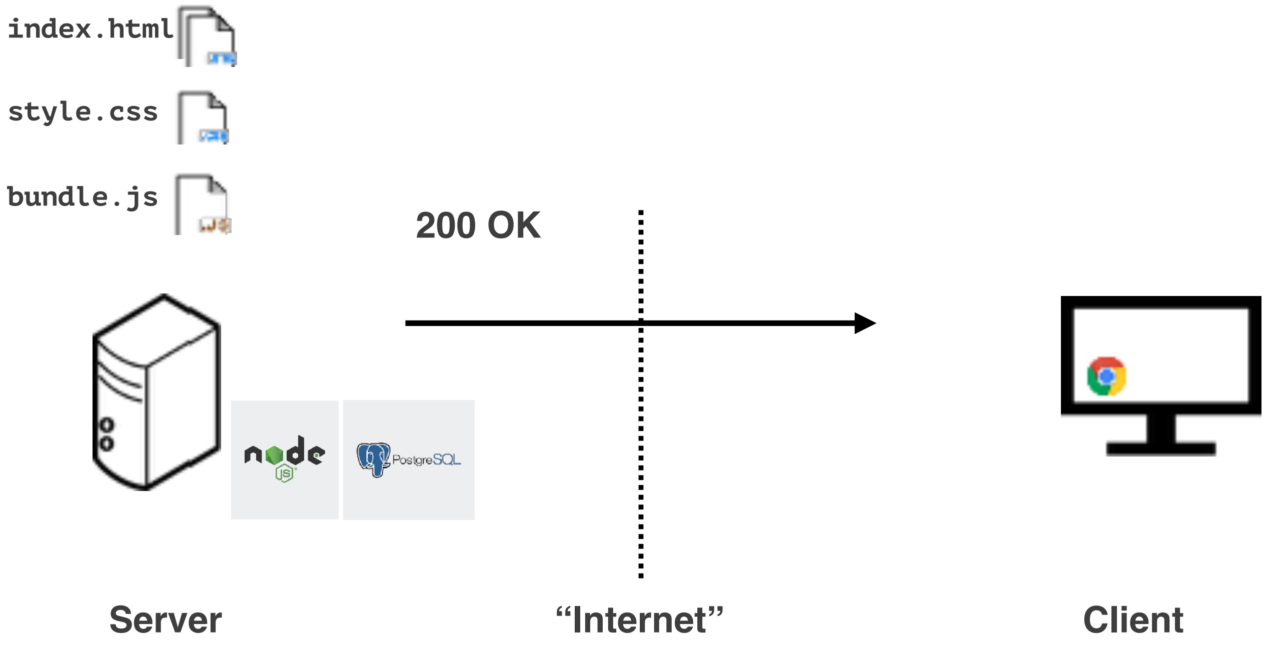
`bundle.js`

HTTP GET /

Server

“Internet”

Client



index.html

style.css

bundle.js

200 OK

Server

“Internet”

Client

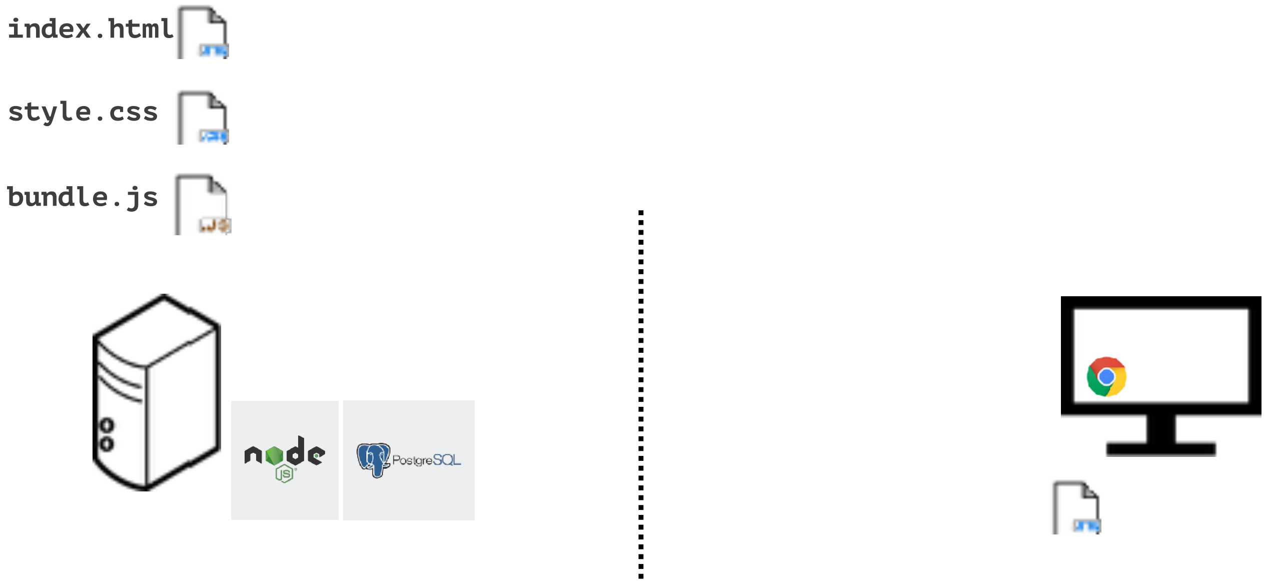
```html
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id=“app"></div>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id=“app"></div>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id=“app"></div>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id=“app"></div>
  </body>
</html>
```

```html
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id=“app"></div>
  </body>
</html>
```



index.html

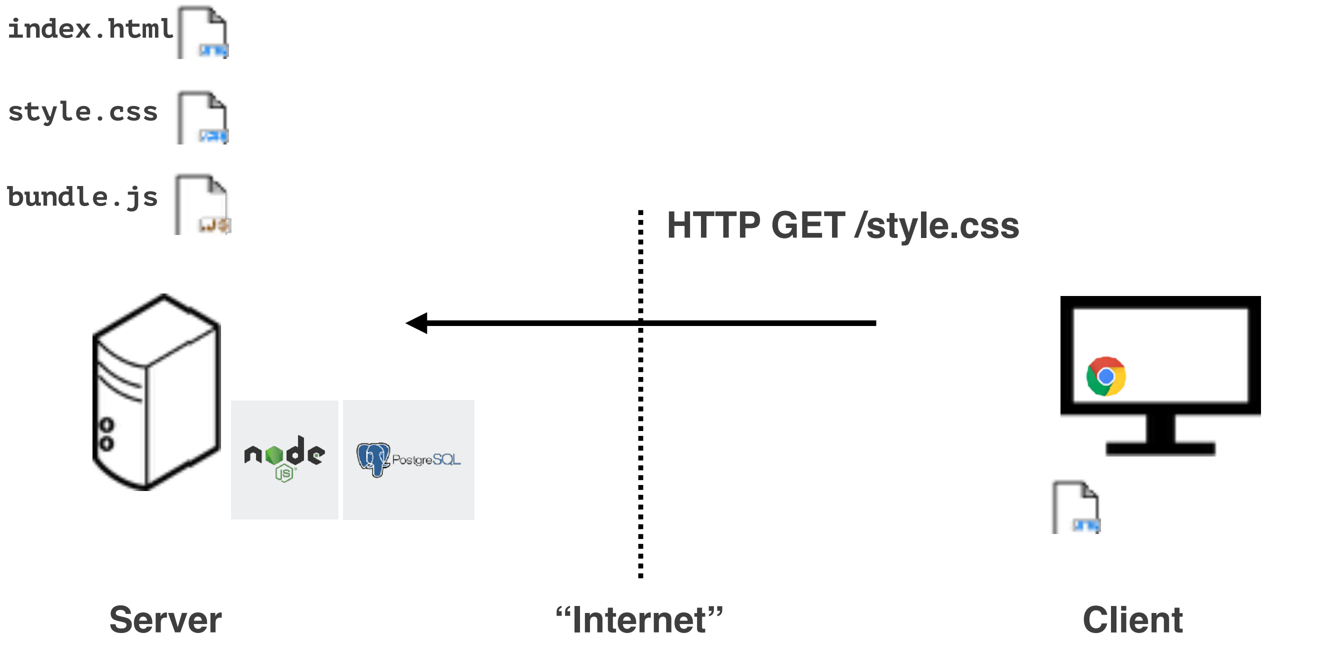
style.css

bundle.js

Server

“Internet”

Client



`index.html`

`style.css`

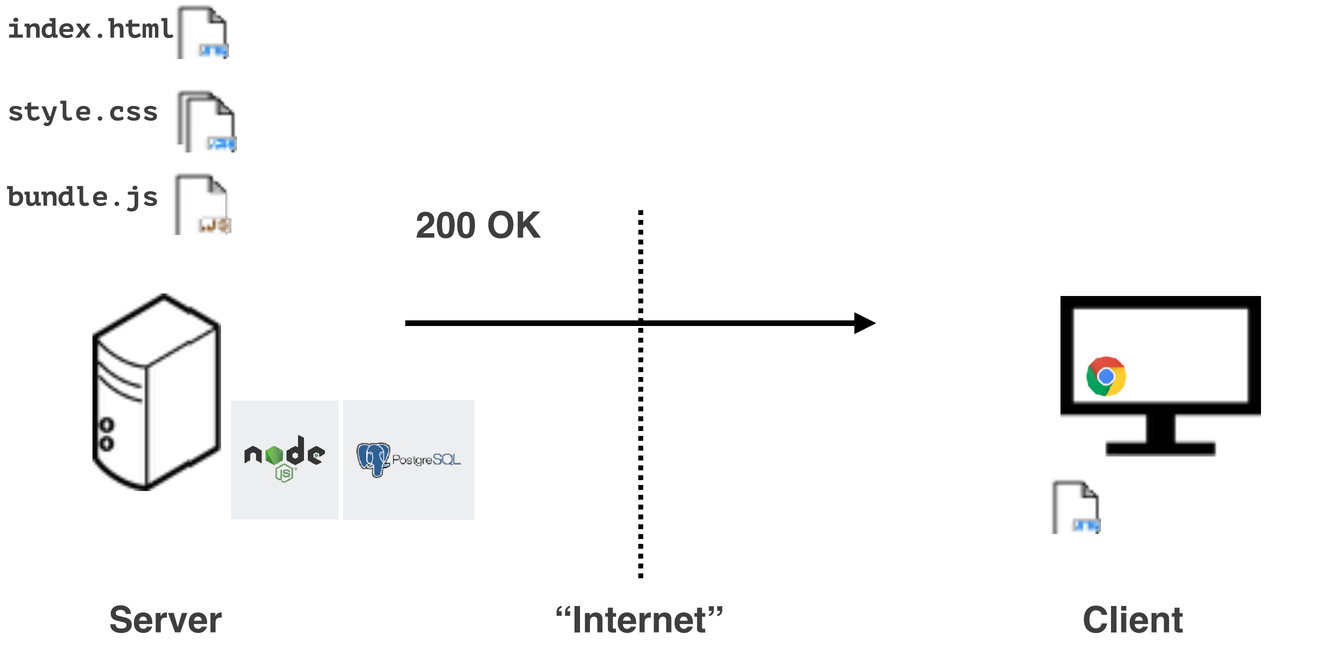
`bundle.js`

HTTP GET /style.css

Server

“Internet”

Client



index.html

style.css

bundle.js

200 OK

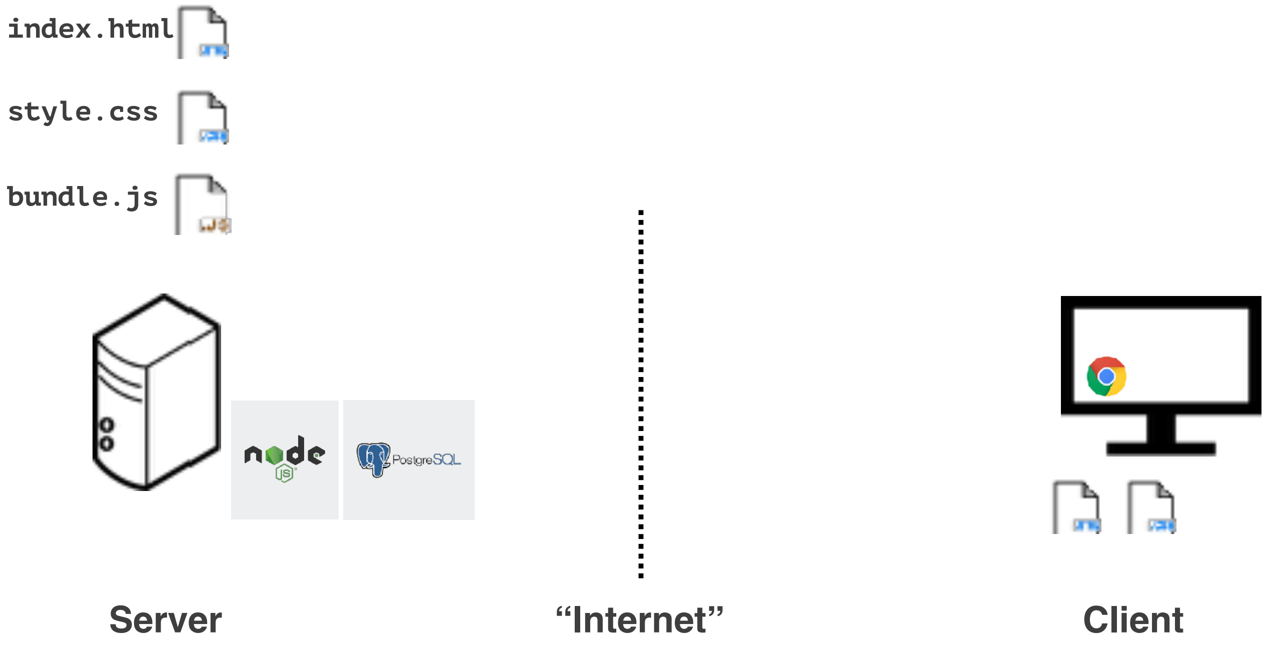
Server

“Internet”

Client

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id=“app"></div>
  </body>
</html>
```

```html
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id=“app"></div>
  </body>
</html>
```



index.html

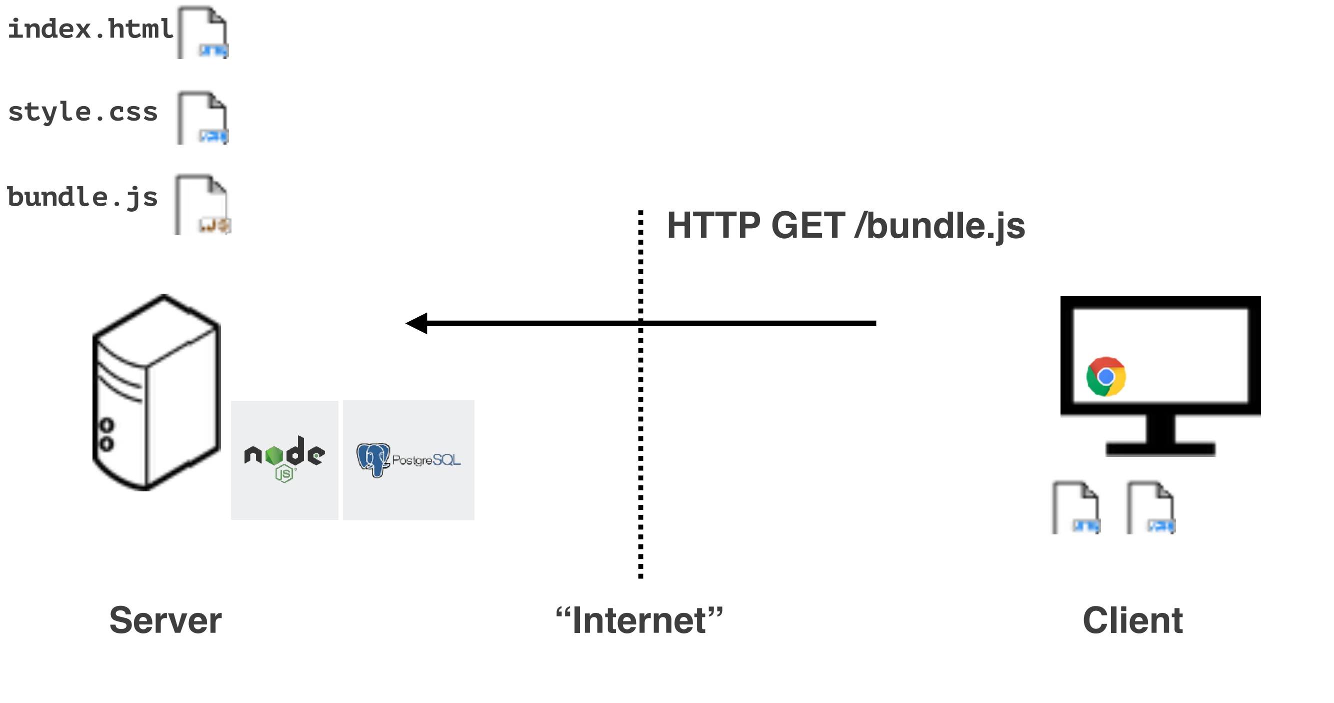
style.css

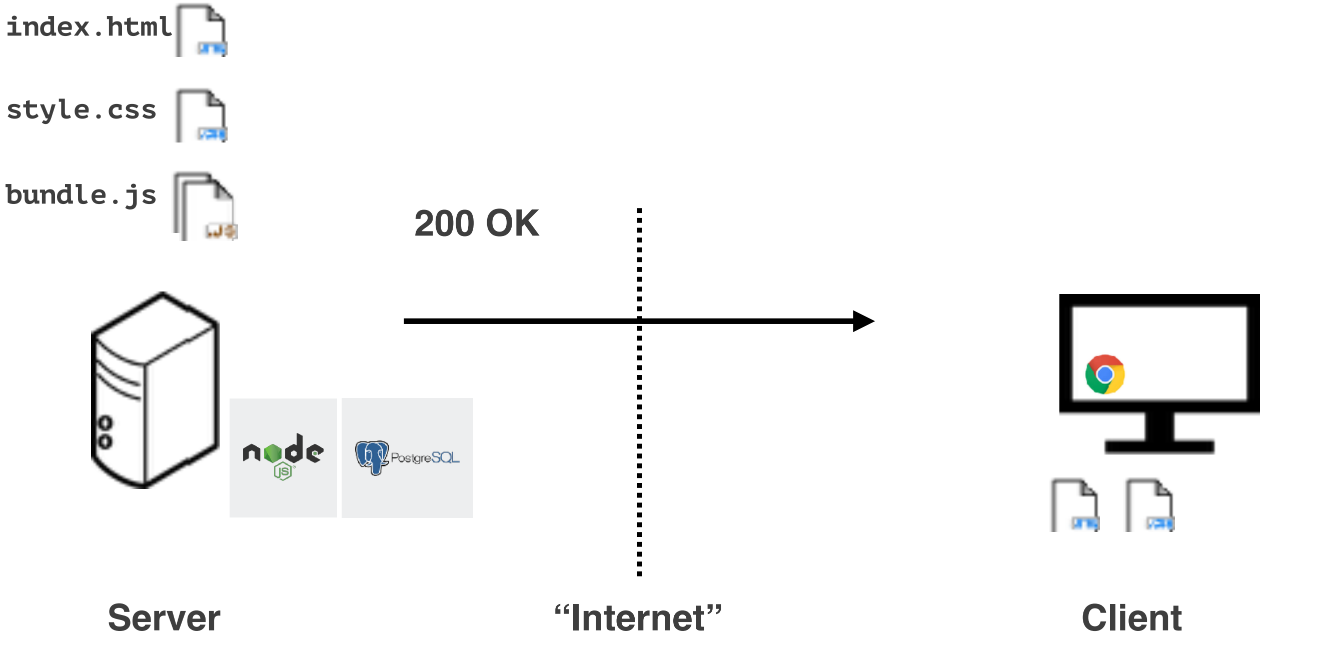
bundle.js

Server

“Internet”

Client


index.html
style.css
bundle.js
HTTP GET /bundle.js
node JS
PostgreSQL
Server
“Internet”
Client



index.html

style.css

bundle.js

200 OK

Server

“Internet”

Client

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id=“app"></div>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id=“app"></div>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id=“app"></div>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id=“app"></div>
  </body>
</html>
```

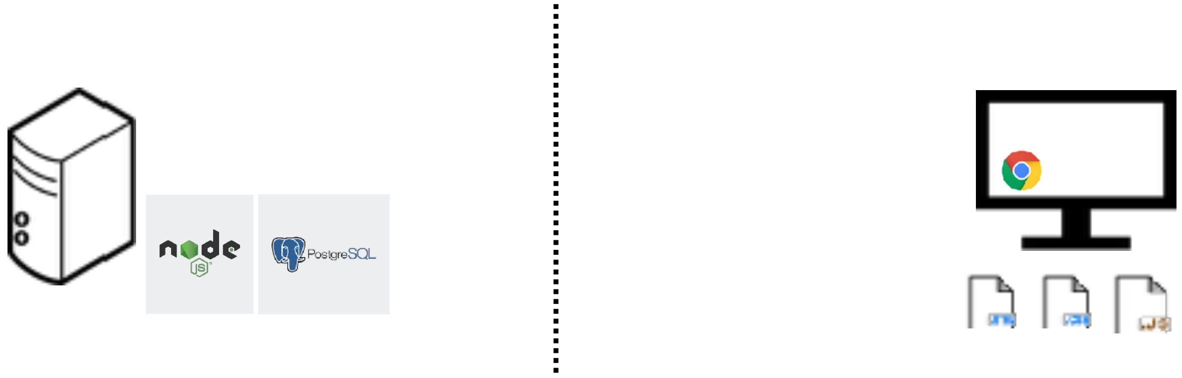
```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id=“app"></div>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id=“app"></div>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id=“app"></div>
  </body>
</html>
```

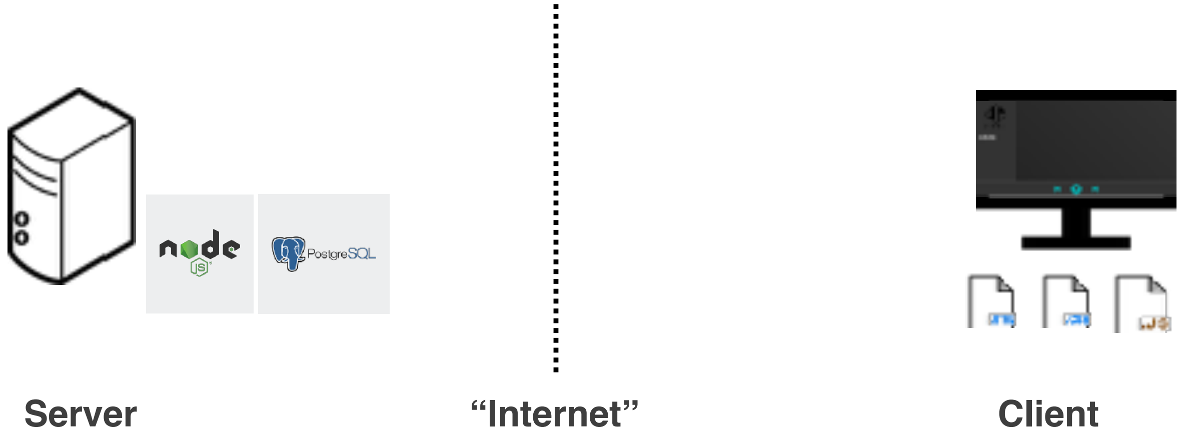
Time to execute this bundle.js file!



Server

“Internet”

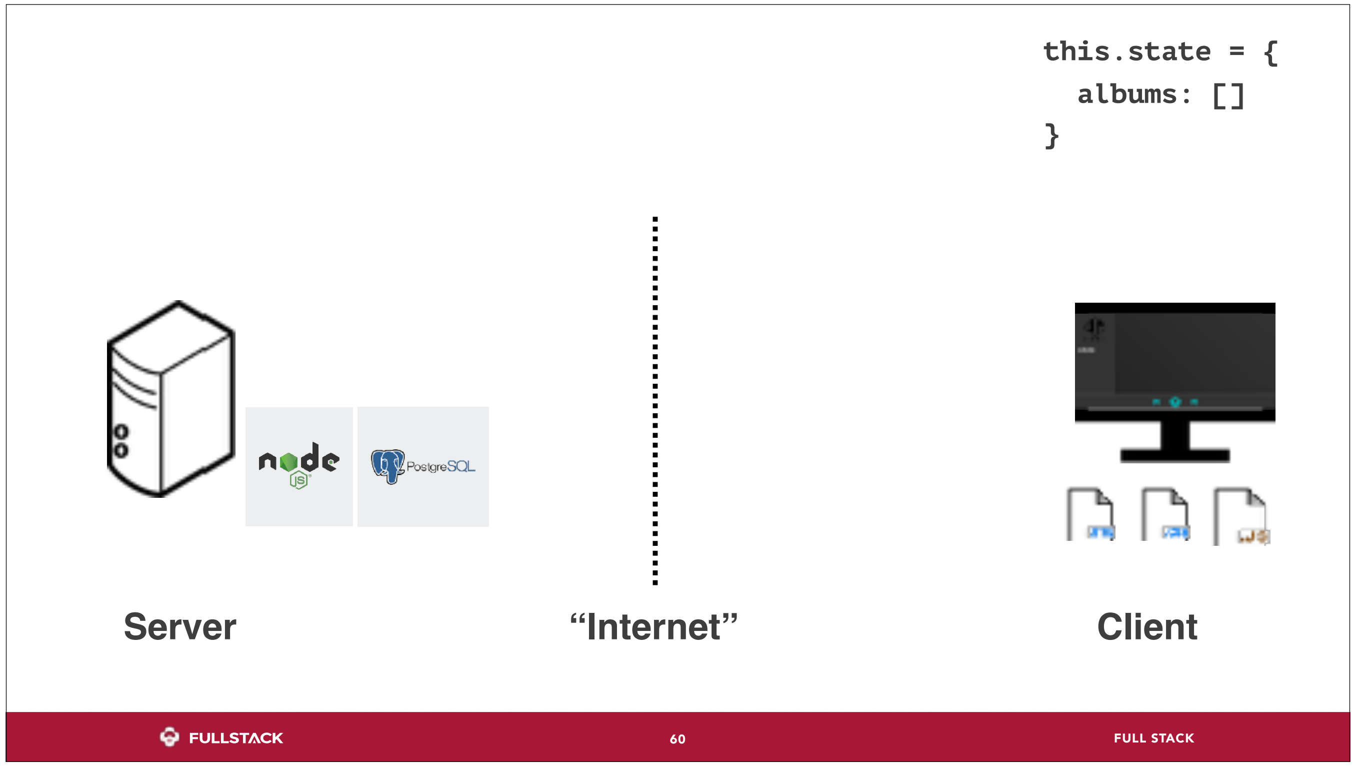
Client



Server

“Internet”

Client



```
this.state = {
  albums: []
}
```

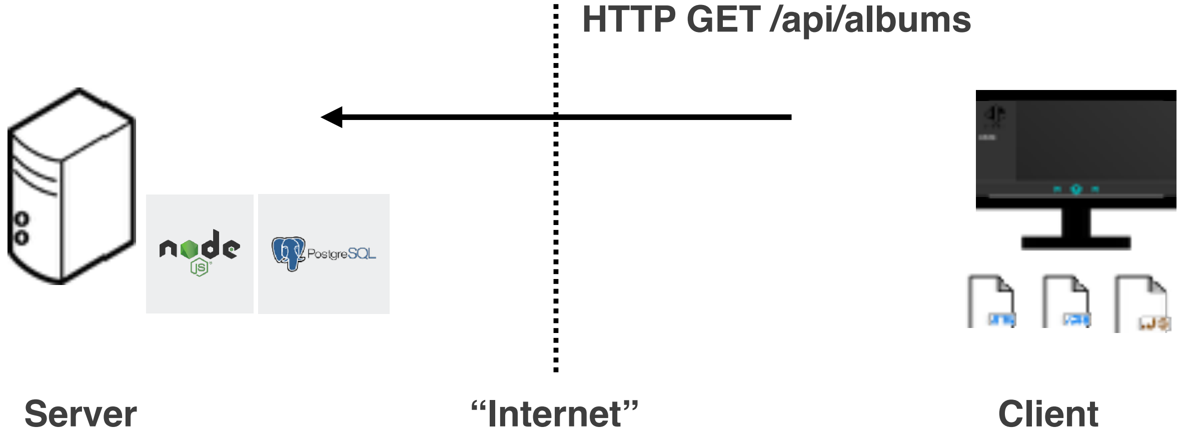
Server

“Internet”

Client

This is our initial render and state, now we can make Ajax calls in componentDidMount

```
this.state = {
  albums: []
}
```



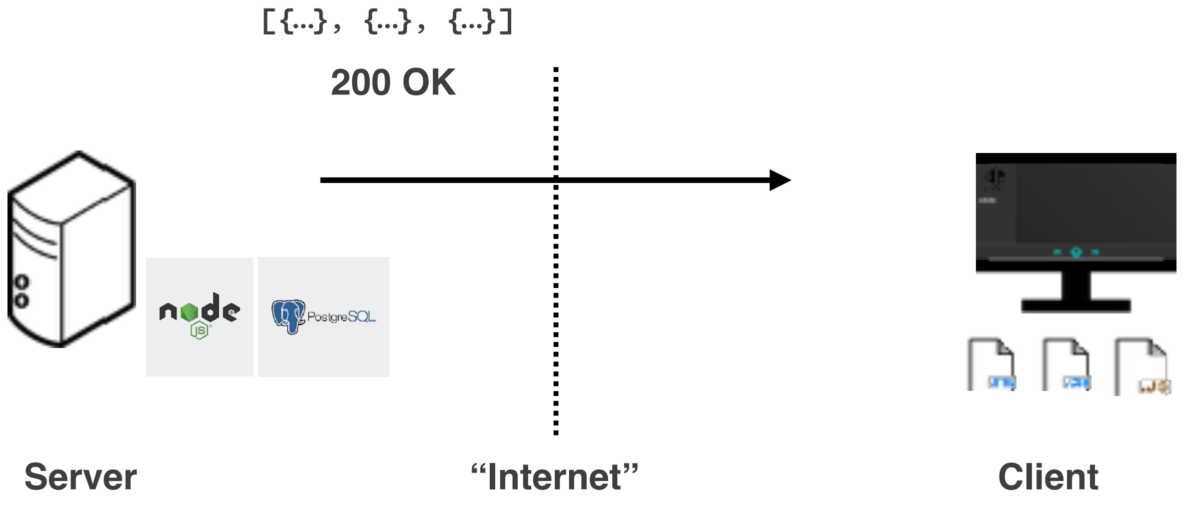
HTTP GET /api/albums

Server

“Internet”

Client

```
this.state = {
  albums: []
}
```



[{…}, {…}, {…}]

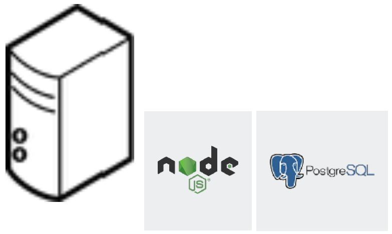
200 OK

Server

“Internet”

Client

```
this.state = {
  albums: [
    {…},
    {…}
  ]
}
```



Server

“Internet”

Client