Introduction to APIs

Introduction to APIs

client-side
React code

API

database

1. **Writing your own API <--- TODAY**
2. Using your own API
3. Using someone else's API

Agenda

How does it fit in with what we already know? (high level)
What is an API?

Preview of API code (very little new stuff!)
REST methods

Insomnia (HTTP client application)

How does it fit in with what we already know? (detailed, optional)
Demo

Look at today's challenge

Agenda

How does it fit in with what we already know? **(high level)**
What is an API?

Preview of API code (very little new stuff!)
REST methods

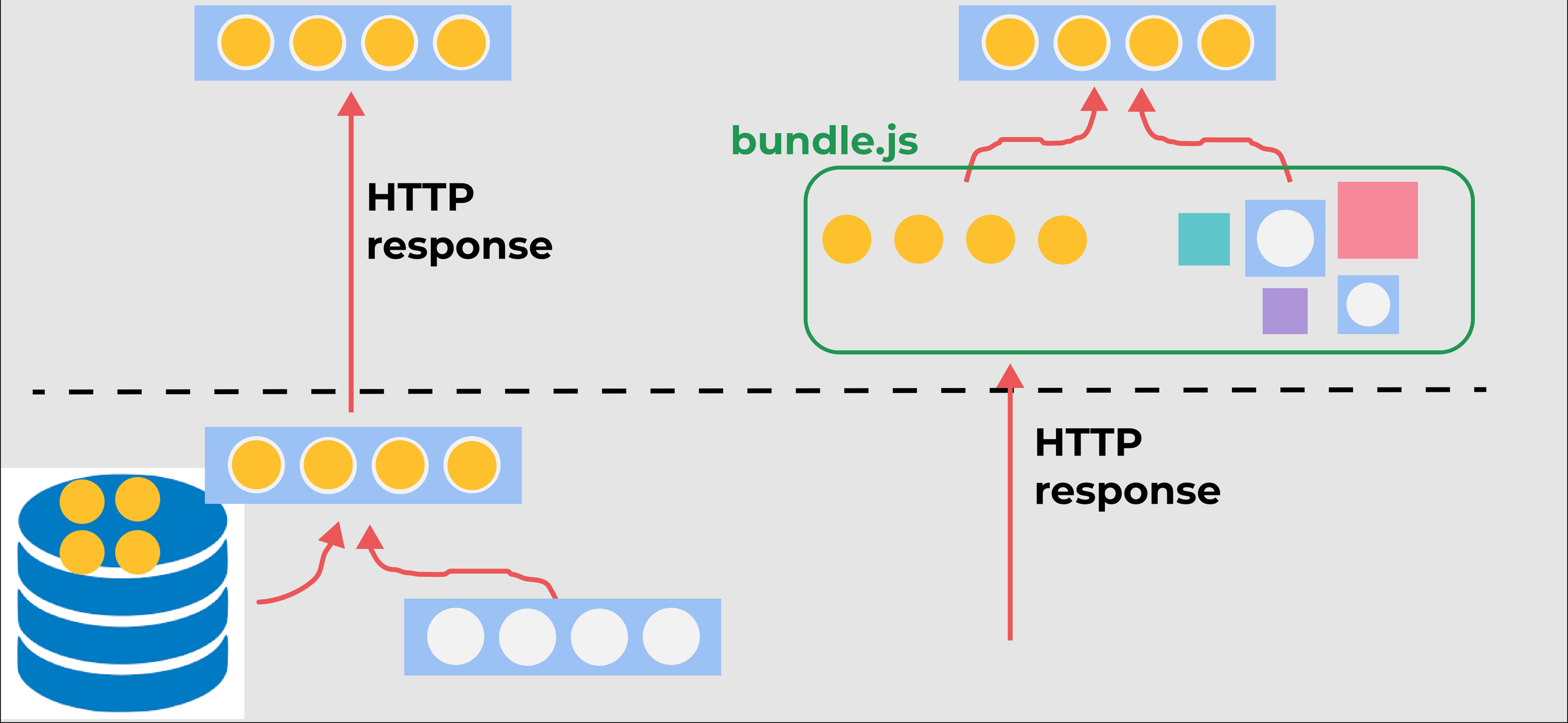
Insomnia (HTTP client application)

How does it fit in with what we already know? (detailed, optional)
Demo

Look at today's challenge

Server-side rendering + database (sprint 3)

Client-side rendering with JSON data (sprint 4)



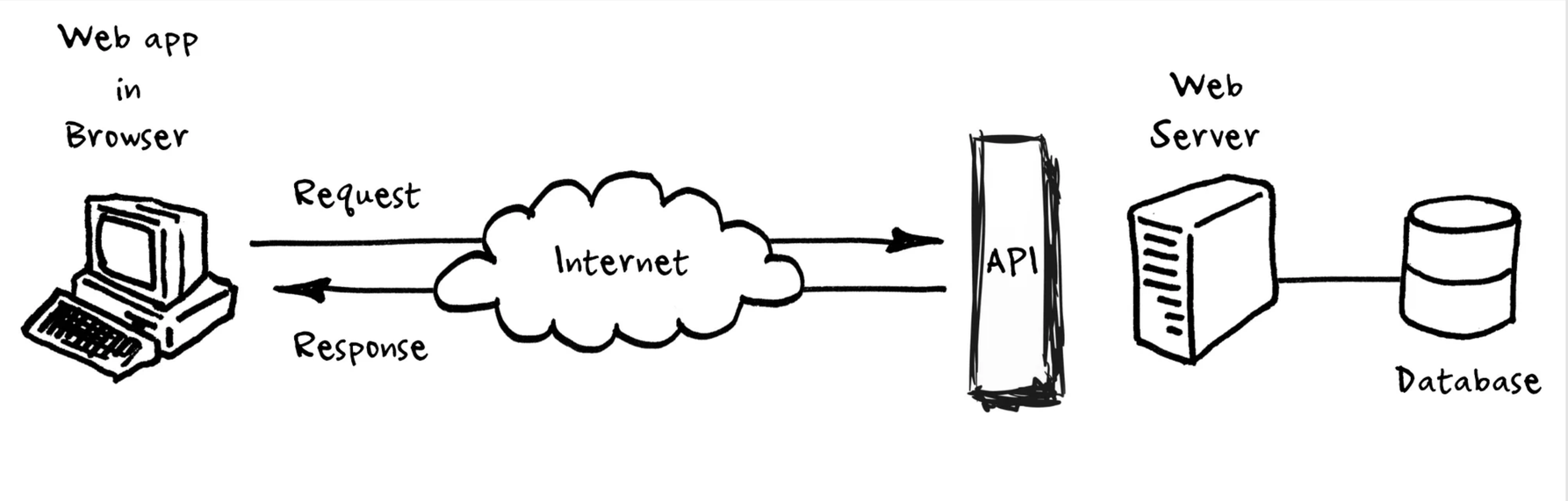
Intro

API = Application Programming Interface

Like a user interface, but for other applications

API is a contract

> “The API is not the database or even the server, it is the code that governs the access point(s) for the server”
>
> - Perry Eising



Web APIs

Web APIs are HTTP for data

REST JSON HTTP APIs
/ RESTful JSON web APIs

REST - REpresentational State Transfer

Agenda

How does it fit in with what we already know? (high level)
What is an API?

Preview of API code (very little new stuff!)
REST methods

Insomnia (HTTP client application)

How does it fit in with what we already know? (detailed, optional)
Demo

Look at today's challenge

Preview: Not much new code today, just concepts and practice...

sending text

```
router.get('/', (req, res) => {
  res.send('Hello World')
})
```

sending HTML

```
router.get('/', async (req, res) => {
  const puppies = await getPuppies()
  res.render('home', puppies)
})
```

sending JSON

```
router.get('/', async (req, res) => {
  const sharks = await db.getSharks()
  res.json(sharks)
})
```

RESTful JSON web APIs

Read:

GET /sharks	returns an array of shark objects
GET /sharks/4	returns a single shark object

Create:

POST /sharks	details in req.body

Update:

PUT /sharks/4	replaces existing object
or **PATCH** /sharks/4	alters existing object

Delete:

DELETE /sharks/4

RESTful JSON web APIs

Read:

GET /sharks

GET /sharks/4

Create:

POST /sharks

Update:

PUT /sharks/4

or **PATCH** /sharks/4

Delete:

DELETE /sharks/4

Existing data

```
{
    “name”: “Baby Shark”,
    “colour”: “yellow”
}
```

Update

```
{
    “name”: “Junior Shark”
}
```

Result - **PUT (replaces)**

```
{
    “name”: “Junior Shark”
}
```

Result - **PATCH (alters)**

```
{
    “name”: “Junior Shark”,
    “colour”: “yellow”
}
```

API versioning

http://localhost:3000/api/**v1**/sharks
http://localhost:3000/api/**v2**/sharks

https://api.ebay.com/post-order/**v2**/return/{returnId}?fieldgroups={groupEnum}

Agenda

How does it fit in with what we already know? (high level)
What is an API?

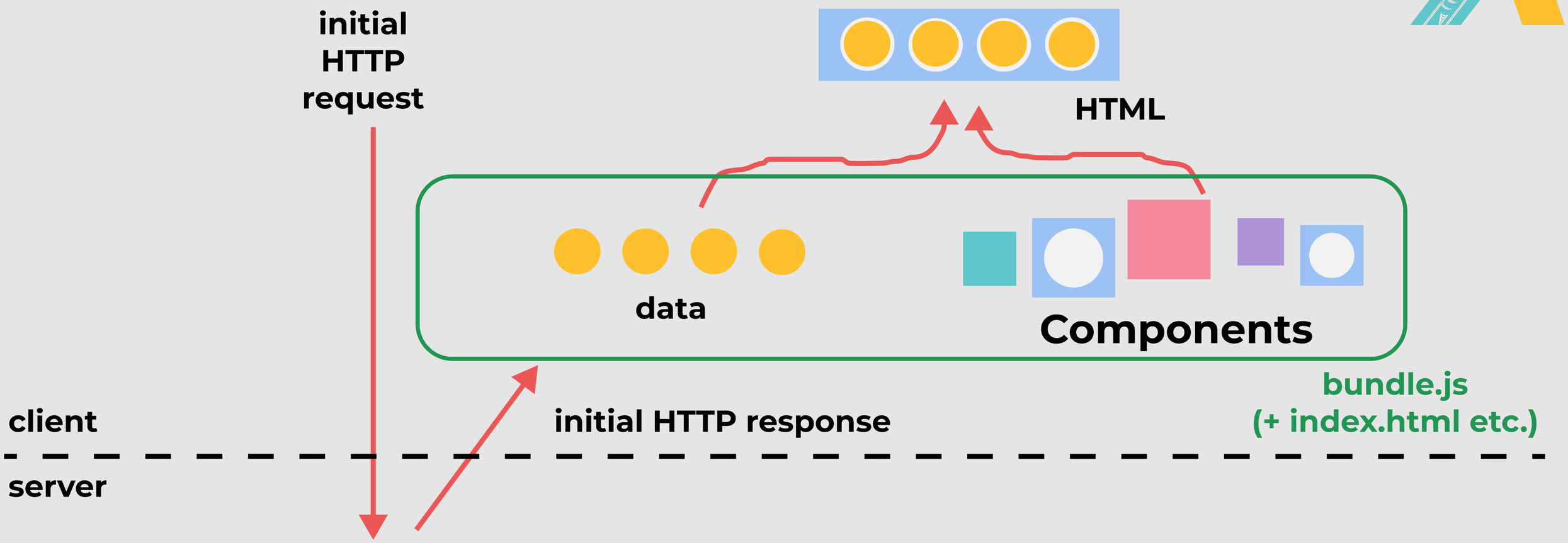
Preview of API code (very little new stuff!)
REST methods

Insomnia (HTTP client application)

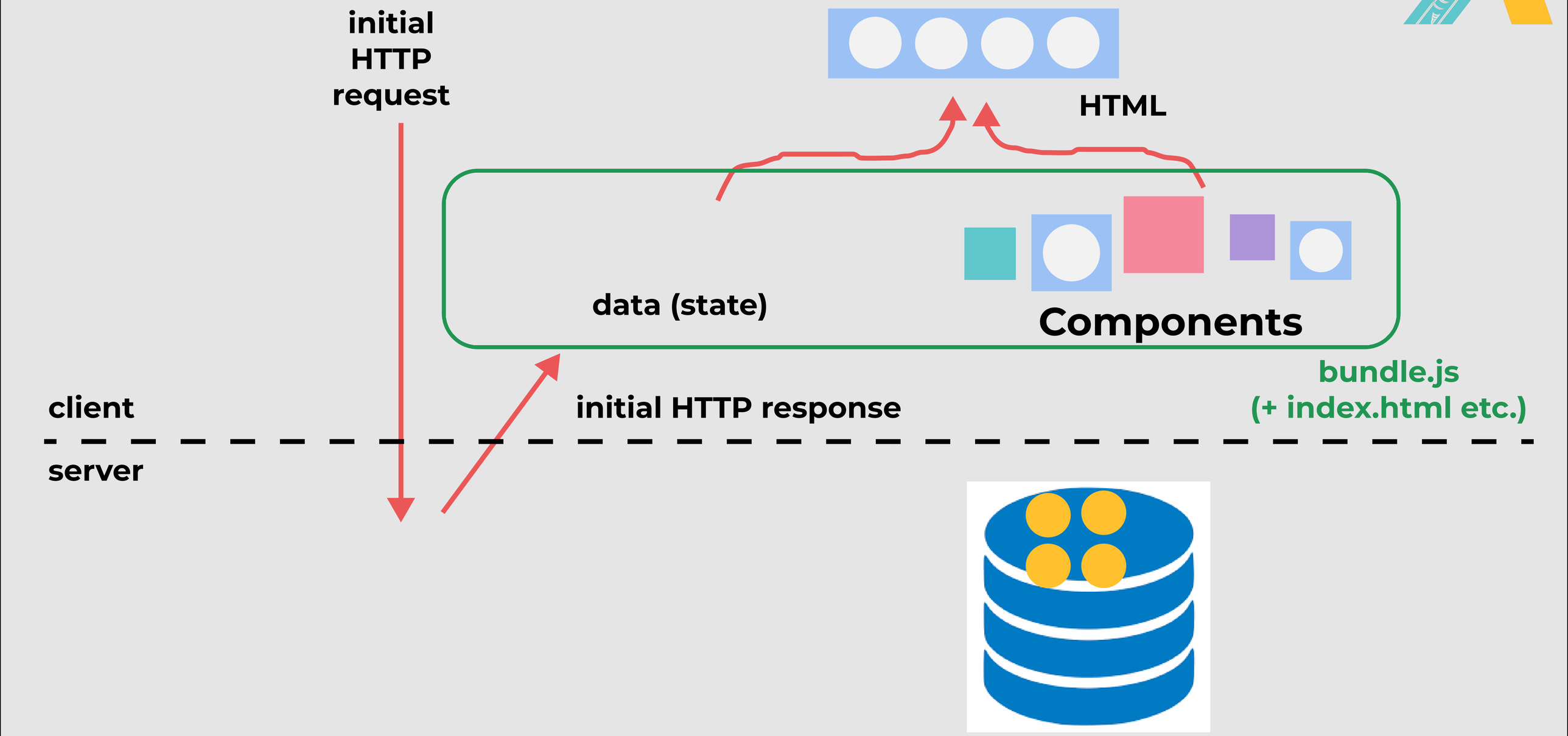
How does it fit in with what we already know? (detailed, optional)
Demo

Look at today's challenge

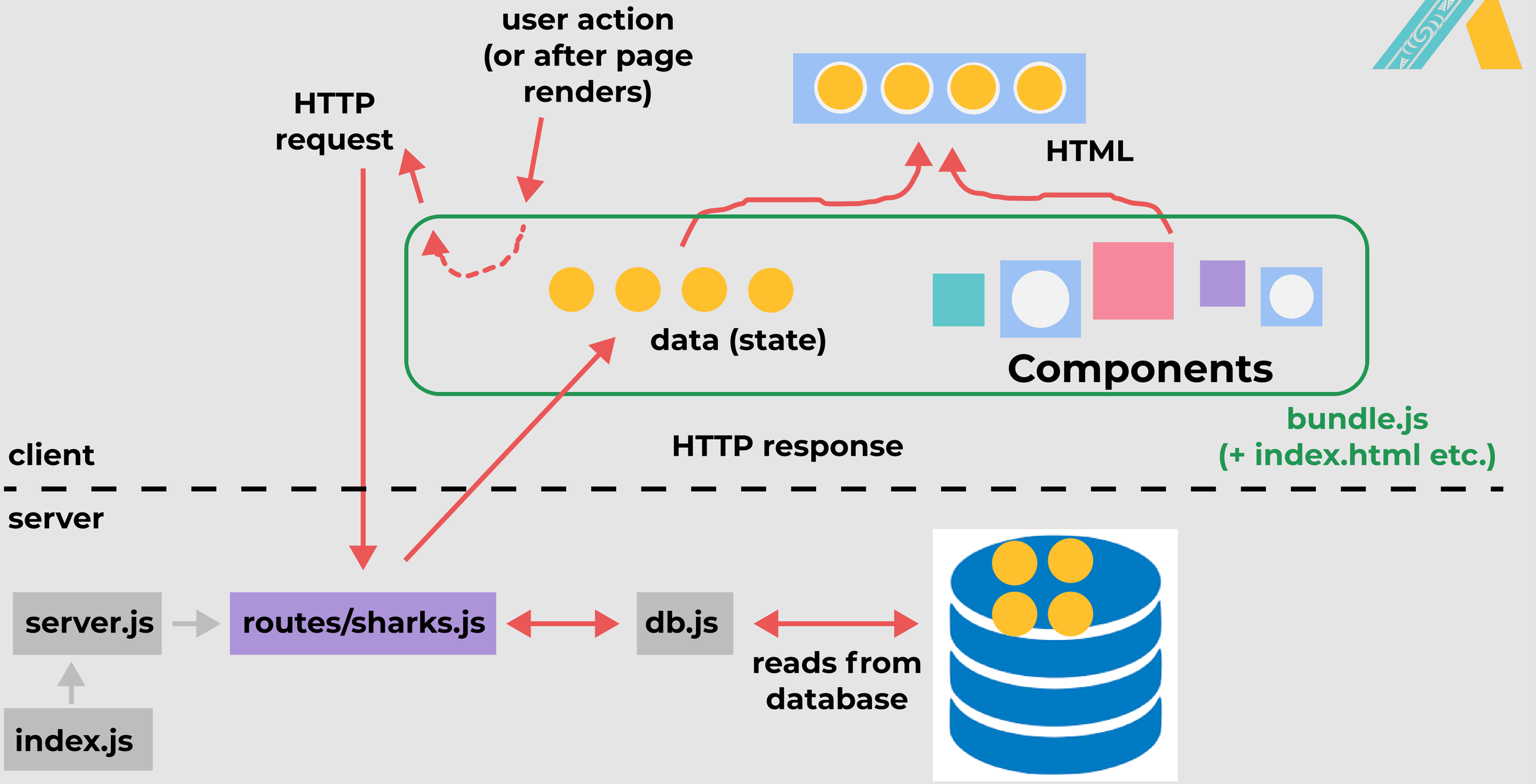
React with client-side JSON data



React with db - initial page load



Read (/GET) request



Read (/GET) request

HTTP request

```
router.get('/', async (req, res) => {
  const sharks = await db.getSharks()
  res.json(sharks)
})
```

data (JSON)

HTTP response

client

server

server.js → routes/sharks.js ↔ db.js ↔ reads from database

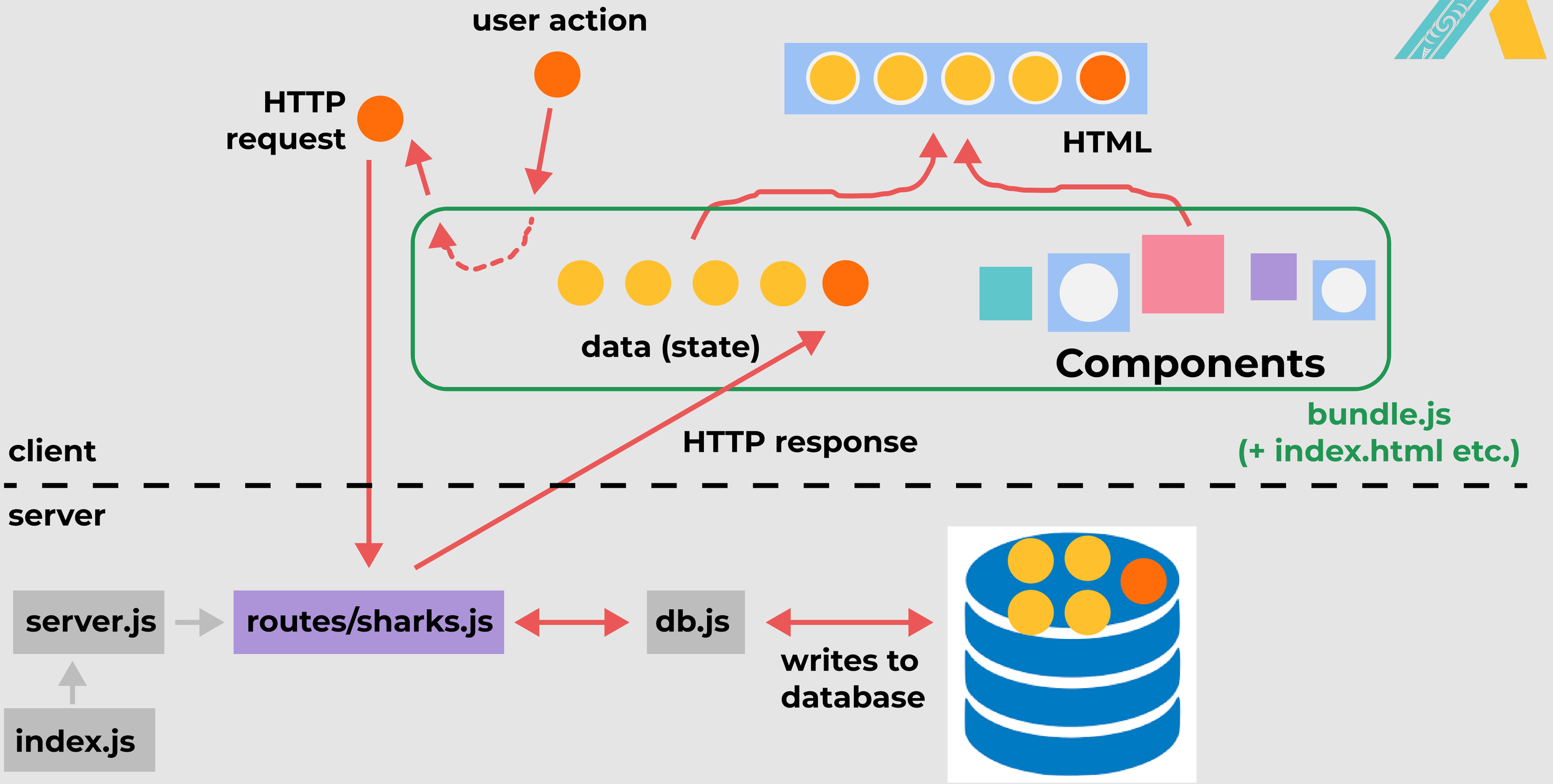
index.js

Code demo

1. Configure server to use your routes
2. Write your routes
3. Write any db functions that your routes need
4. Test it with Insomnia
5. Try it out with the React client-side

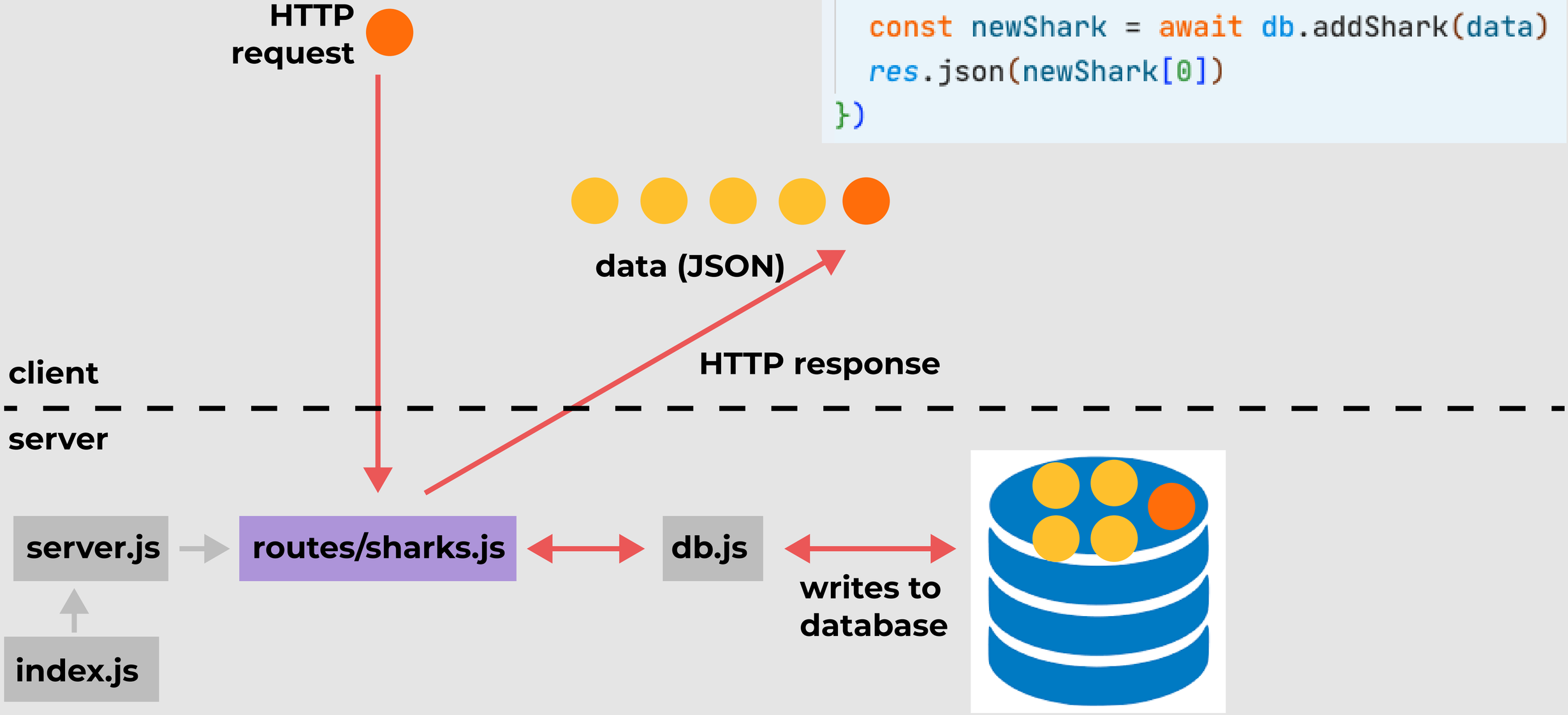
6. Write tests (this afternoon)

Create (/POST) request



Create (/POST) request

```
router.post('/', async (req, res) => {
  const data = req.body
  const newShark = await db.addShark(data)
  res.json(newShark[0])
})
```



Summary

1. Writing your own API **<--- TODAY**
2. Using your own API
3. Using someone else's API

API is like a user interface, but for other applications to use
Governs the access point(s) for the server
API is a contract

GET /sharks	array of shark objects
GET /sharks/4	single shark object
POST /sharks	details in req.body
PUT /sharks/4	replaces existing object
or **PATCH /sharks/4**	alters existing object
DELETE /sharks/4	

How to write API routes and call db functions from them
Insomnia

Agenda

How does it fit in with what we already know? (high level)
What is an API?

Preview of API code (very little new stuff!)
REST methods

Insomnia (HTTP client)

How does it fit in with what we already know? (detailed, optional)
Demo

Look at today's challenge

Gotchas

Converting from snake_case to camelCase

```
export function getSharks(): Promise<SharkFull[]> {
  return db('sharks').select('first_name as firstName')
}
```

If things aren’t working, consider whether you might have inserted bad data in your own database

Please ask us if you get stuck!

Note that the challenge does not closely follow this demo