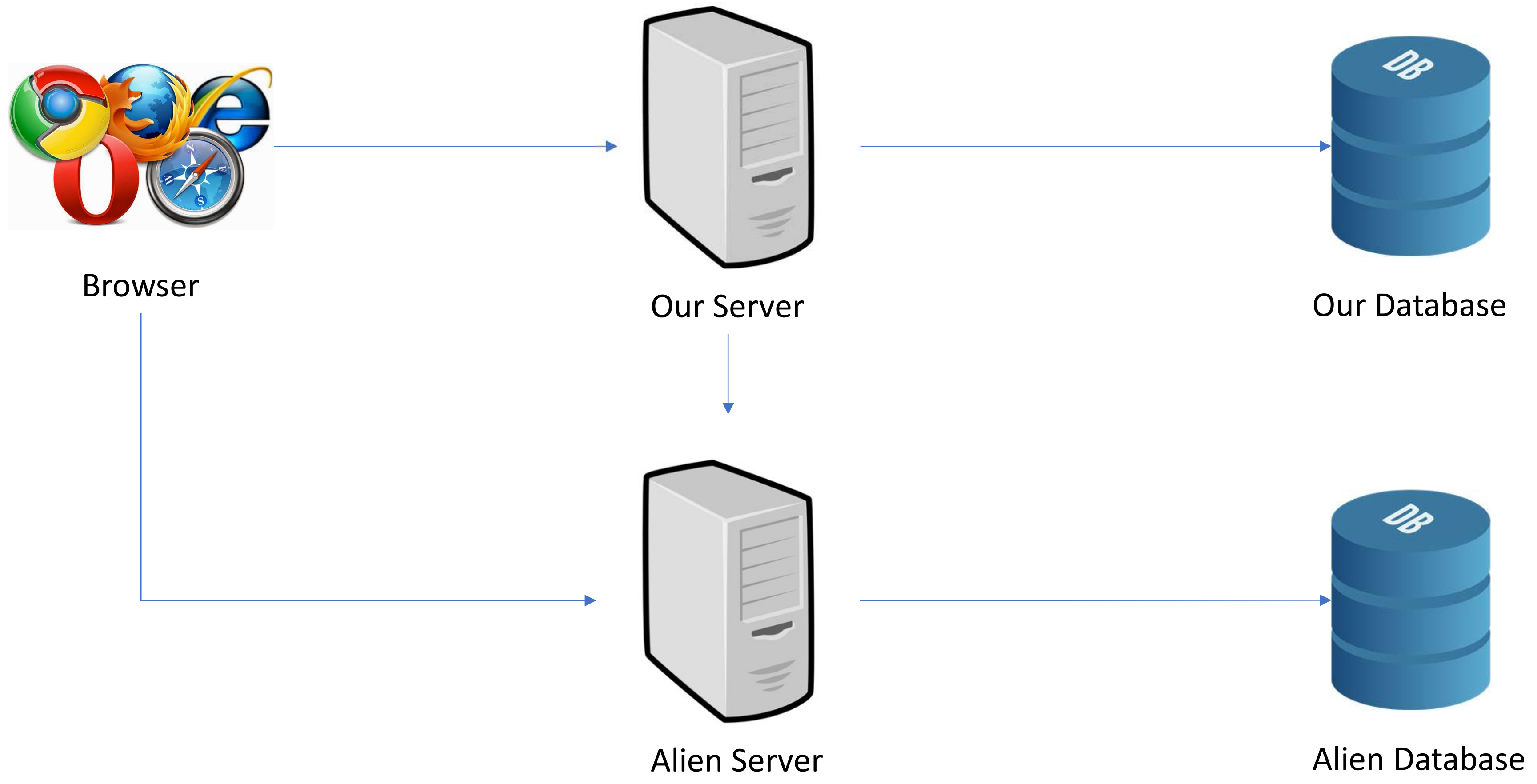
MODULE 3

Fetch and Promises

Changing Architecture?



Web Services and APIs

- Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks.

- An API (Application Programming Interface) is a set of features and rules that exist inside a software program (the application) enabling interaction with it through software

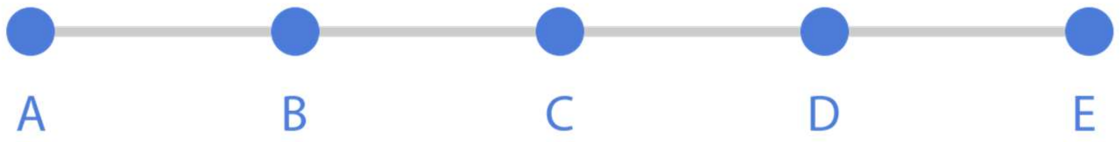
Creating and Consuming

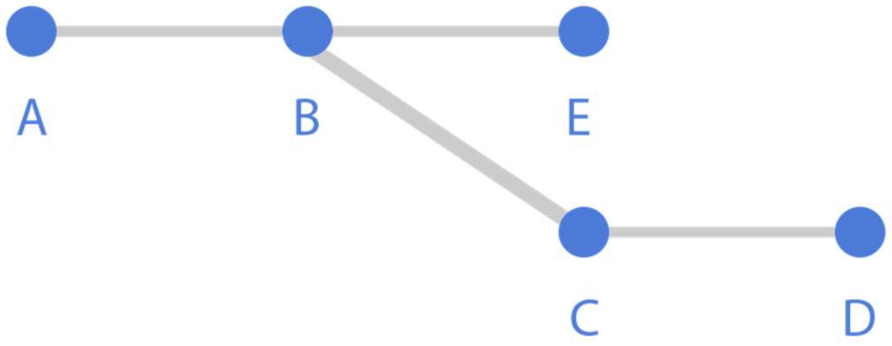
- **Creating** a web service is simply exposing methods and properties from classes
- **Consuming** a web service is calling those APIs and getting the data.

Asynchronous Programming

- Our programs: Synchronous

Asynchronous Programming





Fetch API

- Fetch API provides an interface for accessing and manipulating parts of the HTTP pipeline

- Promise<Response> fetch(input[, init]);

- Input can be
  - A USVString (String) containing the direct URL of the resource you want to fetch.
    - This could be a local resource
    - This could be a remote resource
  - A Request object.

Sample Code

```
fetch(‘https://api.bitbucket.org/2.0/teams?role=member')
   .then( (response) => {
        return response.text();
   })
   .then( (data) => {
      document.getElementById('results').innerHTML = data; }
   });
```

Promises, promises.

- I promise I will return!
- Three states:
  - Pending: initial state, neither fulfilled nor rejected.
  - Fulfilled: meaning that the asynchronous operation completed successfully.
  - Rejected: meaning that the asynchronous operation failed.
- Use .then() to access functions when promise returns
- Use .catch() for errors

LET'S CODE!

ELEVATE YOURSELF

WHAT QUESTIONS DO YOU HAVE?

Reading for tonight:

Introduction to Vue