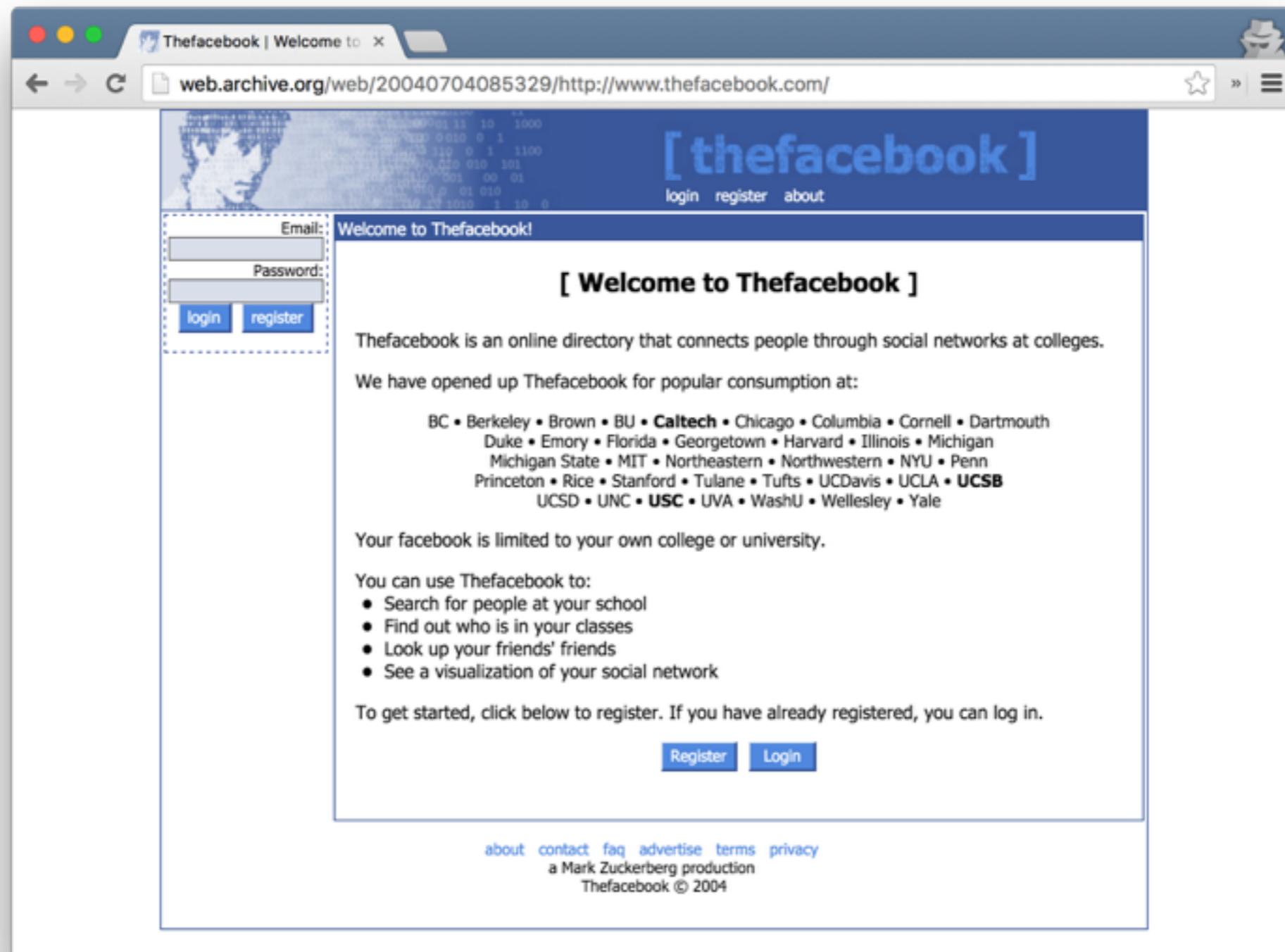


JavaScript Web Applications

Hiun Kim

Computer Science Department, Sejong University
Seoul, South Korea

We are old, the web as we know..



New Waves of Modern Web



Hiun Kim

Edit Profile

FAVORITES

News Feed

20+

Messages

Events

Pressdojo

Close Friends

20+

Saved

6

SUGGESTED

Node.js Korea

2014 세종인 Come ... 12

Divtag

PAGES

Divtag

20+

Pages Feed

Like Pages

Create Page

Create Ad

GROUPS

Node.js Korea

워드프레스 홈페이지 20+

The Hacker News

Closed Group

[Discussion](#)[Members](#)[Events](#)

PINNED POST

**Mohit Kumar**

November 2 at 1:23am · New Delhi

Hey Friends, Guess What? 😊 Today
5th Anniversary.

Heartiest Thanks all for your Enthusiasm,
Sharing, Love ❤️, Time and Efforts as
We wouldn't be here, five years on ar
support us too.... [See More](#)



뉴스

연예

스포츠

웹툰·뽐

리빙푸드

"힘내라 수험생" 응원 쇄도, 수능 코앞 간절한 기도

"취업자수 늘어"…지표와 현실 사이 괴리감, 왜?

선거구 획정 협상 또 결렬…법정시한 넘기나?

아침 최저 5~13도.. '수능 한파' 없... | 수능날 날씨

POLL 수능 전 스트레스, 잠재우는 나만의 방법은?

실시간급상승

1 설련화

↑339



저녁 뉴스 생중계 | 다시보기



MBN

연합뉴스TV



YTN

KBS



SBS

A CHANNEL

"미얀마 대통령, 평화적 정권 이양 약속"

돌아온 '버블세븐' 반값세븐 과거 되풀이?

독일 '물지마 난민 수용' 폐기...포용서 통제로 나坦



수능 문답지 옮기
관들



15

정치

생활



병원 가져더니...
노인 버리고 간

웹으로 즐기는
만화, 목요웹툰

Single Page Application

Web site that fits on a single web page with the goal of providing a more fluid UX akin to desktop apps

- via Wikipedia

Single Page Application (cont.)

Concepts established circa 2002 (not sure)

US Patents 8136109

(Delivery of data and formatting information to allow client-side manipulation)

Single Page Application (cont.)

Provides layout and partial view

Every event is related to JavaScript

They against the web we know!

Enhancing User Experience

Provides seamless experience through,

Reducing round tripping of submission

Reducing process of CSS & JavaScript

Continuity

Page does not reload completely,
browser can store state with JavaScript

Continuity (cont)

Web is not only a document viewer,

but data manipulator (Google Spread Sheets)

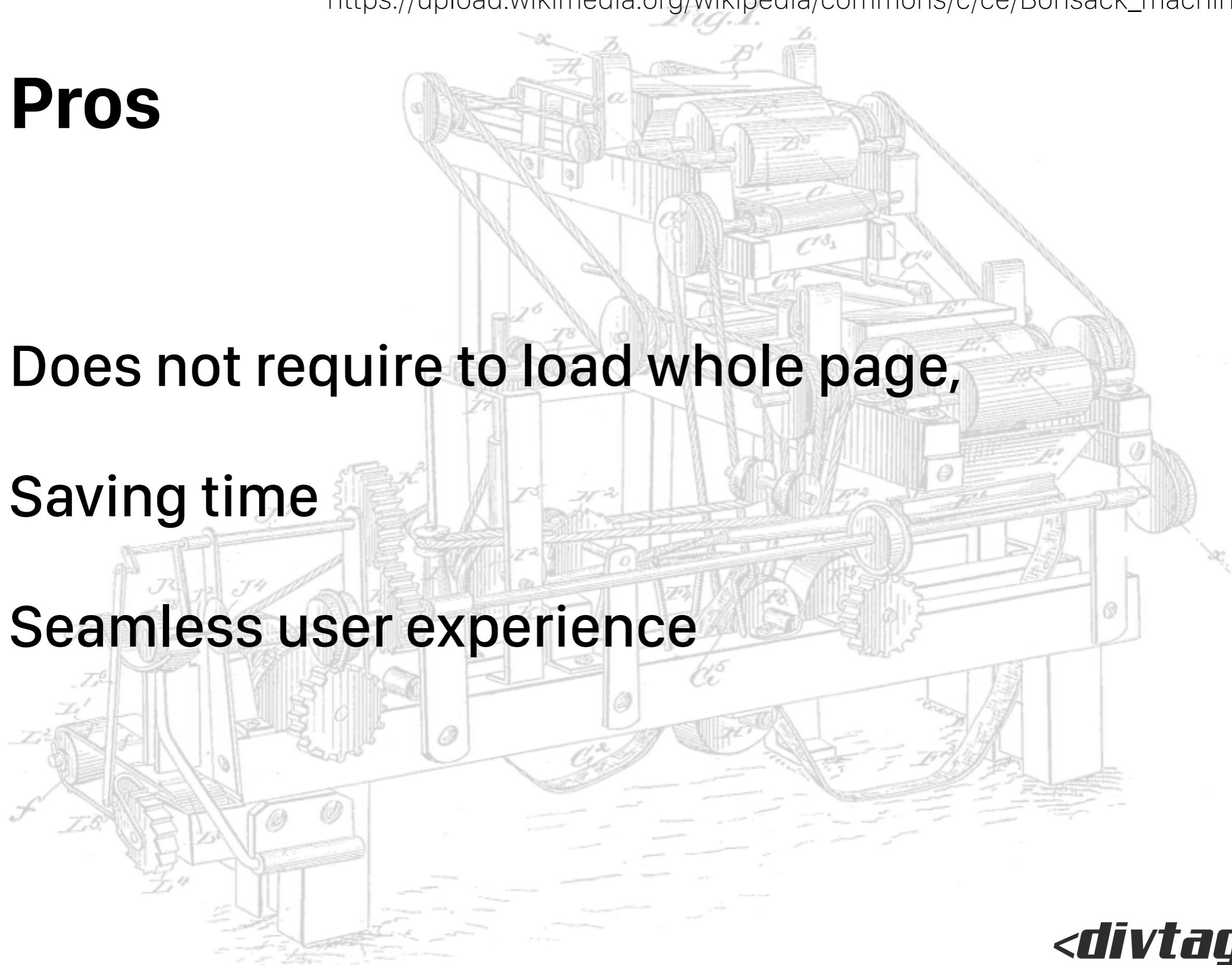
or interactive viewer (Facebook.com)

Pros

Does not require to load whole page,

Saving time

Seamless user experience



Cons

Not simple, not pure (as definition of today)

Often slow for processing

Fallback exist but incomplete

Sometime impossible

Core Mechanism

Browser is sandbox

JavaScript can manipulate URL and History

pushState for creating history

Libraries and Frameworks

jquery-pjax (pushState + ajax)

angularjs

ember.js

etc.

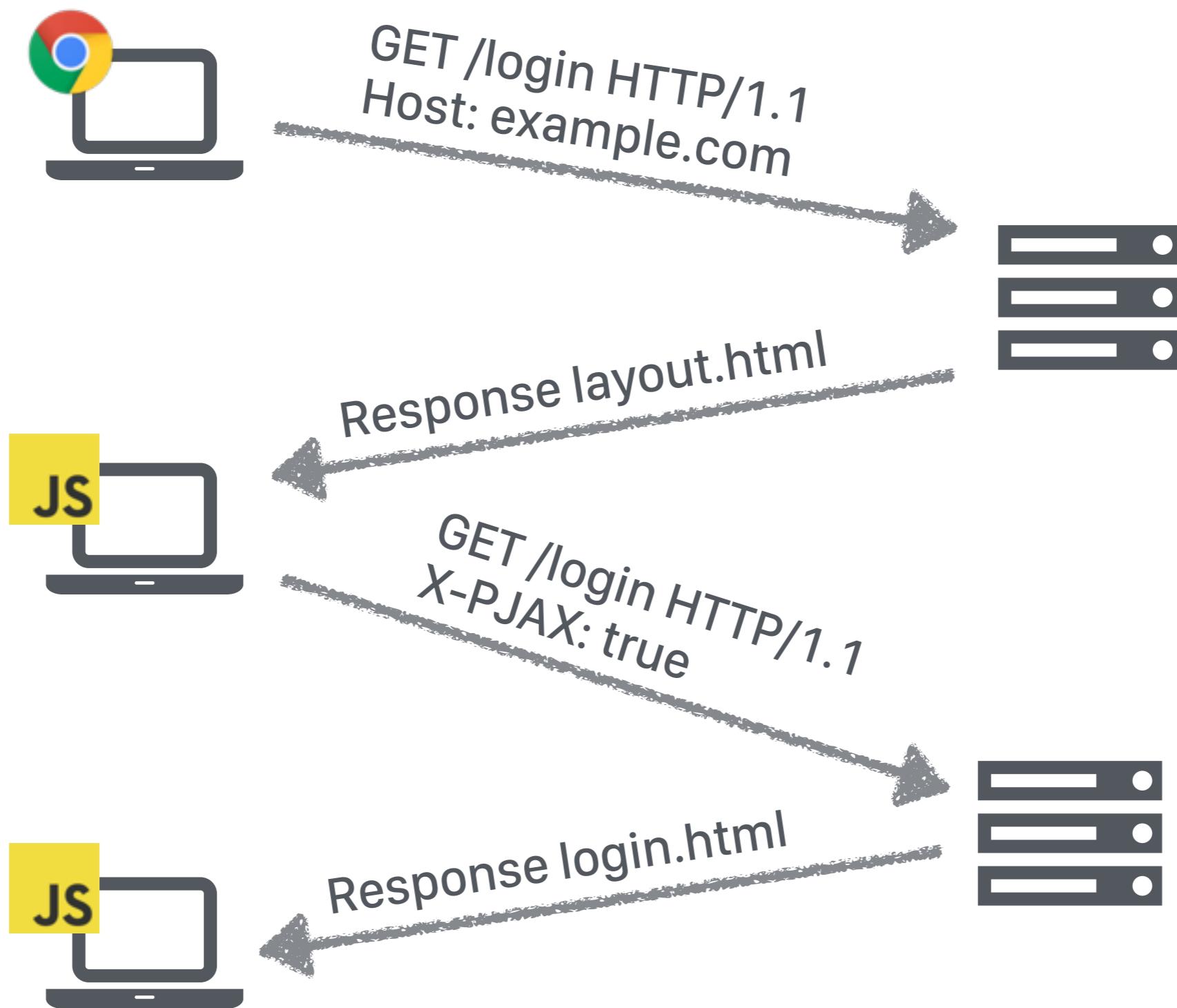
jquery-pjax

```
<!DOCTYPE html>
<html>
<head>
<script= "./jquery.pjax.js"></script>
<script>
$(document).pjax('a', '#pjax-container')
</script>
</head>
<body>
<h1>My Site</h1>
<div class="container" id="pjax-container">
  Go to <a href="/page/2">next page</a>.
</div>
</body>
</html>
```

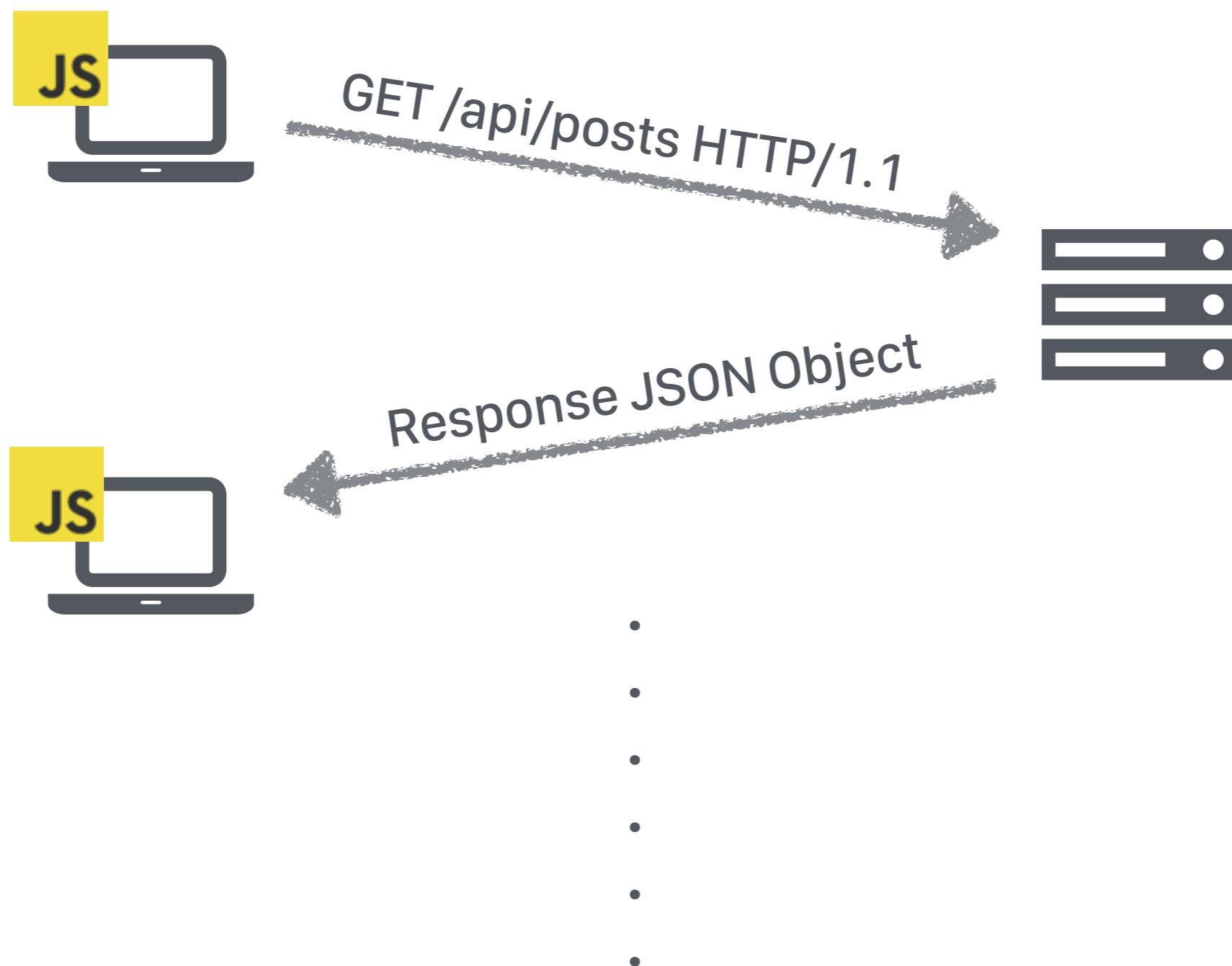
jquery-pjax (cont.)

```
GET /?_pjax=%23main HTTP/1.1
Host: pjax.herokuapp.com
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_1) Ap...
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept: text/html, */*; q=0.01
X-Requested-With: XMLHttpRequest
X-PJAX: true
X-PJAX-Container: #main
Referer: http://pjax.herokuapp.com/dinosaurs.html
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,ko;q=0.6
```

Workflow



Workflow (cont.)



Paradigm shifts in rendering

In usual, logic within template executed

Data is rendered in server

```
<?php  
//PHP renders data directly to template  
for ($x = 0; $x <= 10; $x++) {  
    echo "<b>The number is: $x </b><br>";  
}  
?>
```

Paradigm shifts in rendering (cont.)

In usual, logic within template executed

Server only render template

Client JS renders data

Frameworks



Angularjs
Bidirectional UI Data Binding

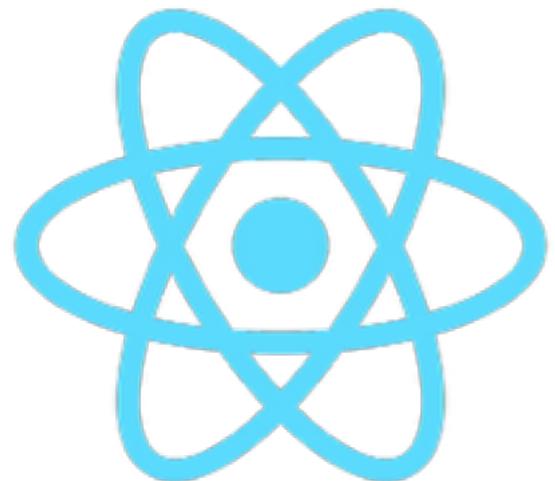


Ember.js
MVC Architectural Pattern



METEOR
Fullstack Framework for SPAs.

Libraries



React
A JavaScript library for
building user interfaces



Vue.js
Reactive Components for
Modern Web Interfaces

New Perspectives

Facebook: Rethink established best practices™

- Ben Alman

XHP

Augments the syntax of the language such that XML document fragments become valid PHP expressions.

XHP (cont.)

```
<?hh  
$href = 'http://www.facebook.com';  
echo <a href={$href}>Facebook</a>;
```

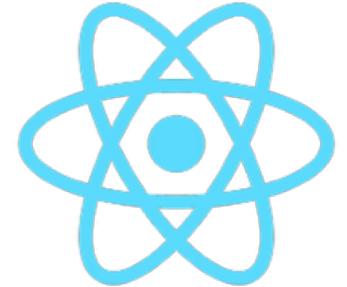
XHP (cont.)

```
<?hh
$post =
<div class="post">
  <h2>{$post}</h2>
  <p><span>Hey there.</span></p>
  <a href={$like_link}>Like</a>
</div>;
```

XHP (cont.)

```
<?hh
$list = <ul />;
foreach ($items as $item) {
    $list->appendChild(<li>{$item}</li>);
}
```

React

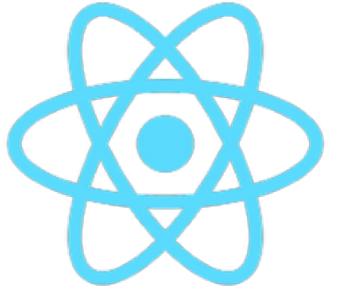


React is a JavaScript library for creating user interfaces by Facebook and Instagram. Many people choose to think of React as the **V** in **MVC**.

We built React to solve one problem: **building large applications with data that changes over time.**

- via facebook.github.io/react

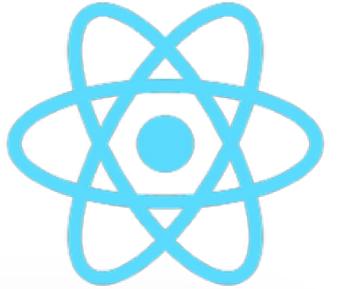
React (cont.)



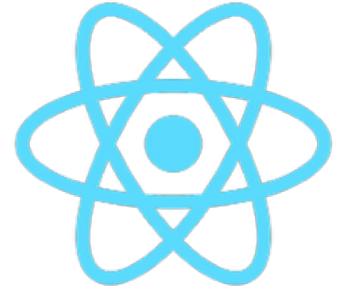
```
var HelloMessage = React.createClass({  
  render: function() {  
    return (  
      <div>  
        <p>Hello {this.props.name + '!'}</p>  
        <i>By React</i>  
      </div>  
    );  
  }  
});
```

```
var mountNode = jQuery('#main')[0];  
ReactDOM.render(<HelloMessage name="Hiun" />, mountNode);
```

React (cont.)

A screenshot of a web browser window titled "react-hiun.html". The address bar shows "file:///Users/hiun/Desktop/react-hiun.html". The page content displays the text "Hello Hiun!" followed by "By React". Below the browser window, the corresponding React component code is shown:

```
<div data-reactid=".9">
  <p data-reactid=".9.0"><span data-reactid=".9.0.0">Hello </span>
  <span data-reactid=".9.0.1">Hiun!</span>
  </p><i data-reactid=".9.1">By React</i>
</div>
```

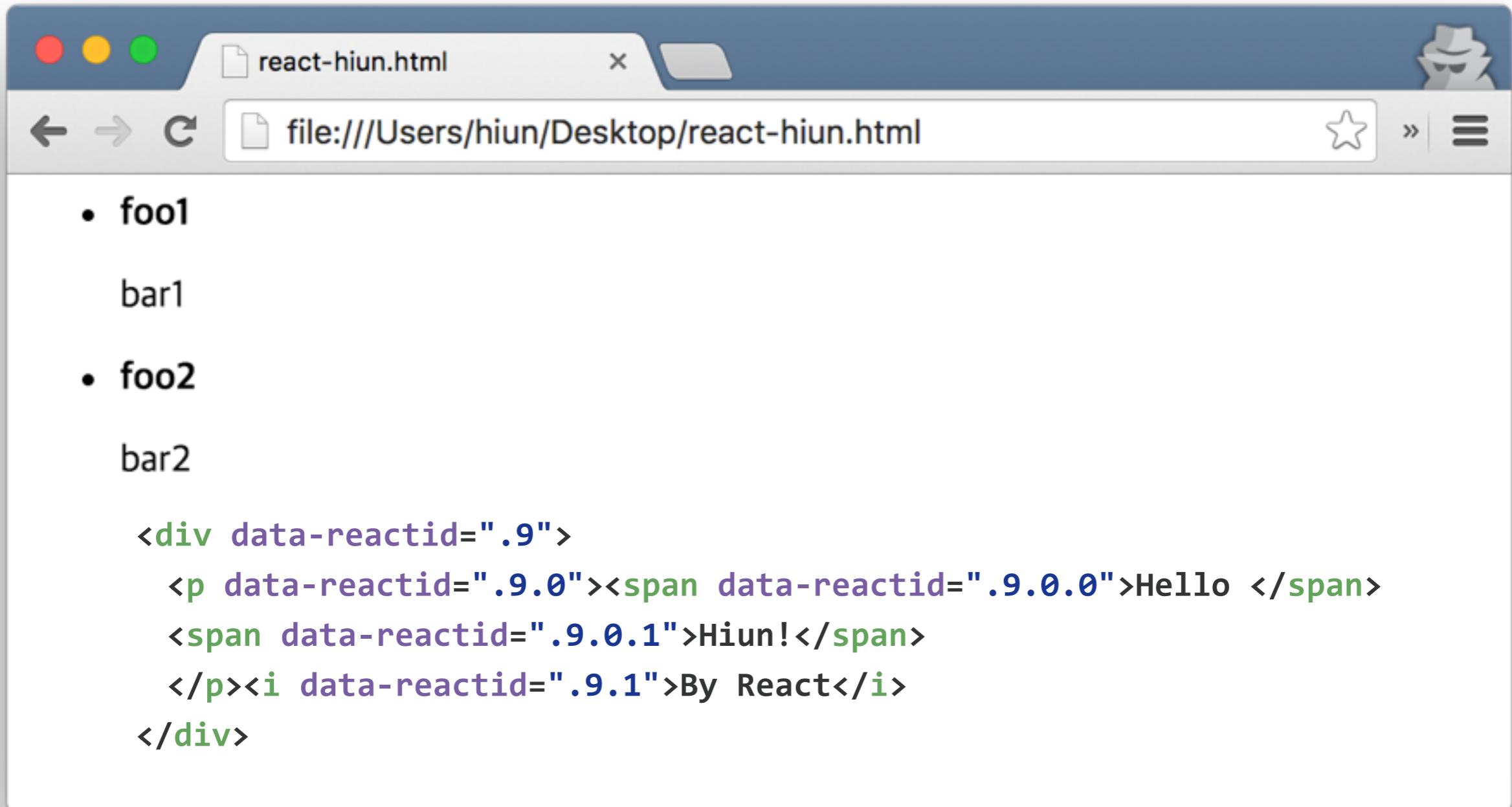
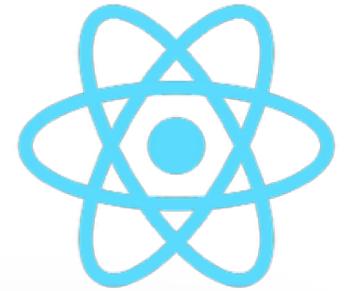


React (cont.)

```
var PostList = React.createClass({
  render: function() {
    return (
      <ul>
        { this.props.posts.map(function (post) {
          return (<li>
            <b>{ post.t }</b>
            <p>{ post.b }</p>
            </li>
          )) }
      </ul>
    );
  }
});
```

```
var posts = [{t: 'foo1', b: 'bar1'}, {t: 'foo2', b: 'bar2'}]
ReactDOM.render(<PostList posts={ posts } />, mountNode);
```

React (cont.)



A screenshot of a web browser window titled "react-hiun.html". The address bar shows "file:///Users/hiun/Desktop/react-hiun.html". The content area displays a list of items and some raw React code. The list includes:

- foo1
 - bar1
- foo2
 - bar2

Below the list is a block of raw React code:

```
<div data-reactid=".9">
  <p data-reactid=".9.0"><span data-reactid=".9.0.0">Hello </span>
  <span data-reactid=".9.0.1">Hiun!</span>
  </p><i data-reactid=".9.1">By React</i>
</div>
```

JSX

A.K.A JavaScript Syntax Extension

These stuff, right there ----->

```
var PostList = React.createClass({  
  render: function() {  
    return (  
      <ul>  
        { this.props.posts.map(function(post) {  
          return (<li>  
            <b>{ post.t }</b>  
            <p>{ post.b }</p>  
            </li>)  
        }))  
      </ul>  
    );  
  }  
});
```

Module

Basic unit of abstraction

Basic unit of managing complexity

ES2015 - Module System

Commonly referred to as ES6,

Provides module system

But preprocessing is required to run in the browser

ES2015 - Module System (cont.)

```
//circle.js
export function areaCalc (r) {
  const PI = Math.PI;
  return PI * r * r;
}

//proc.js
import areaCalc from circle;
areaCalc(3); //28.274333882308138
```

ES2015 - Module System (cont.)

```
import jQuery from 'lib/jquery.js';
import React from 'lib/react.js';
```

```
import autosize from 'lib/autosize.min.js';
import Spinner from 'lib/spin.min.js';
```

```
import validator from 'lib/validator.min.js';
import * as utils from 'utils/index.js';
```

```
import * as post from 'post.js';
```

Bundling the Web

A big code base needs to be organized

A lot of preprocessing is required - SCSS, ES2015

Module systems offer the option to split your code base into modules.

- via webpack.github.io

The Future?

HTML6 proposal -

There's a standard design pattern emerging via all the front-end javascript frameworks where content is loaded dynamically via JSON APIs.

<https://lists.w3.org/Archives/Public/public-whatwg-archive/2015Mar/0071.html>

Thank you

for thinking modern web with us!

slides are available at <http://divtag.sejong.edu>