The Conference for Java
& Software Innovation
Oct 10 – 12, 2016 | London

Chris Bailey | IBM
@Chris_Bailey

Emerging Web Application Architectures
With Java and Node.js

STSM, IBM Runtime Development

developerWorks Open

@Chris__Bailey
@seabaylea

Chris Bailey
IBM Runtime Technologies

The Annual "Death of Java"

Developer trend No. 2: Java's decline as a language will accelerate

The Annual "Death of Java"

Developer trend No. 2: Java's decline as a language will accelerate

Java is in decline … *look at the jobs*. Yes, there are more of them … doing maintenance.

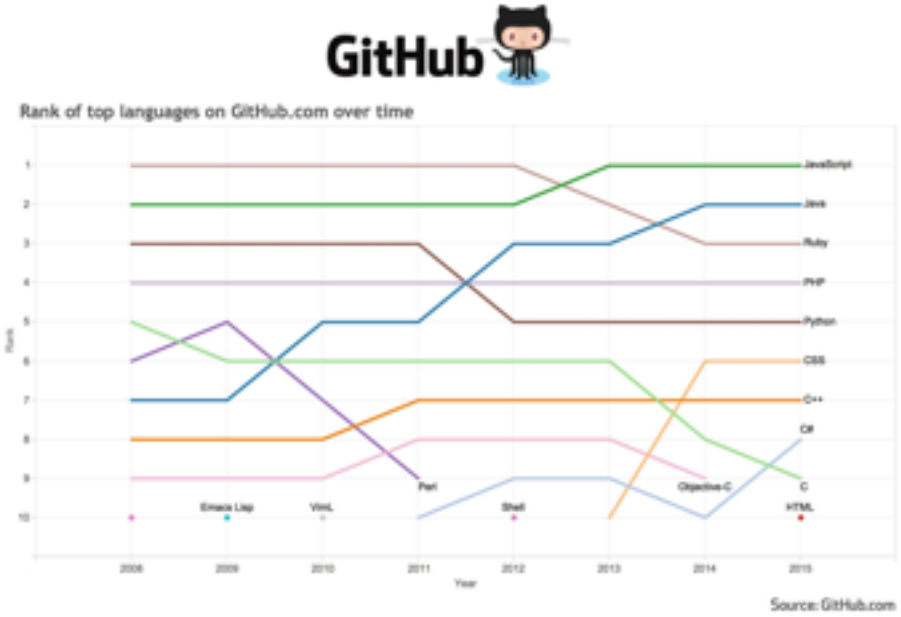
The Annual "Death of Java"

Developer trend No. 2: Java's decline as a language will accelerate

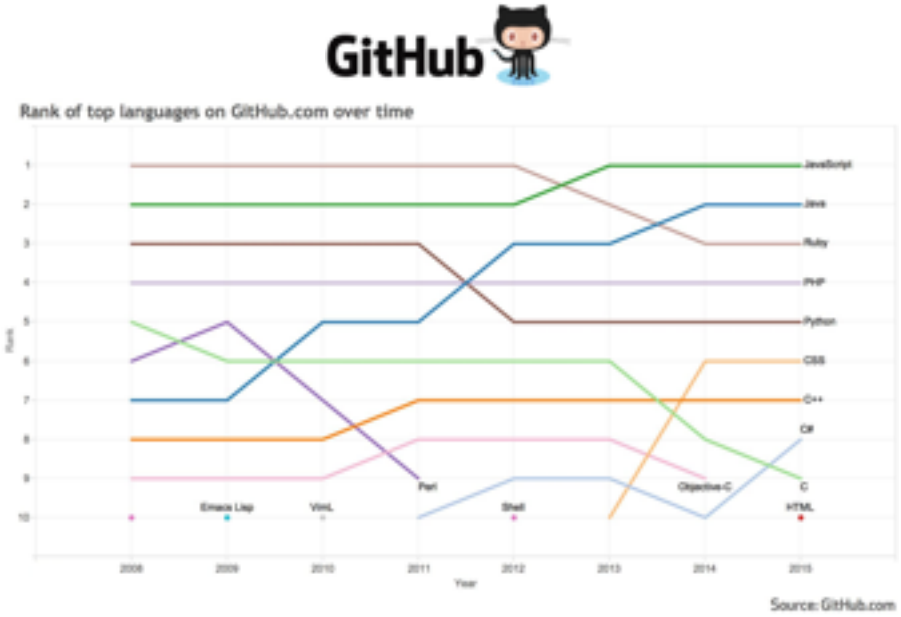
Java is in decline … *look at the jobs*. Yes, there are more of them … doing maintenance.

Now look at the Node.js or Spark or MongoDB job postings.
Those are about doing new development

Developer Ecosystem



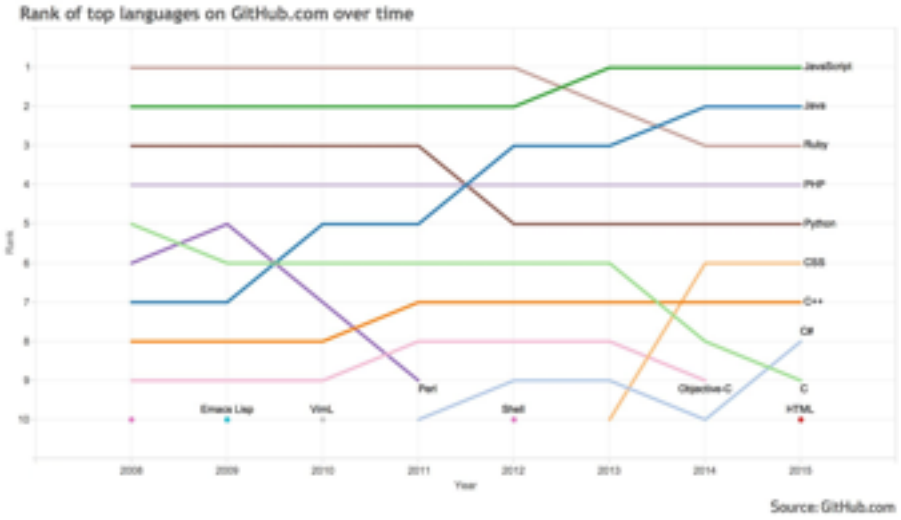
Developer Ecosystem



#1 JavaScript
#2 Java

Developer Ecosystem

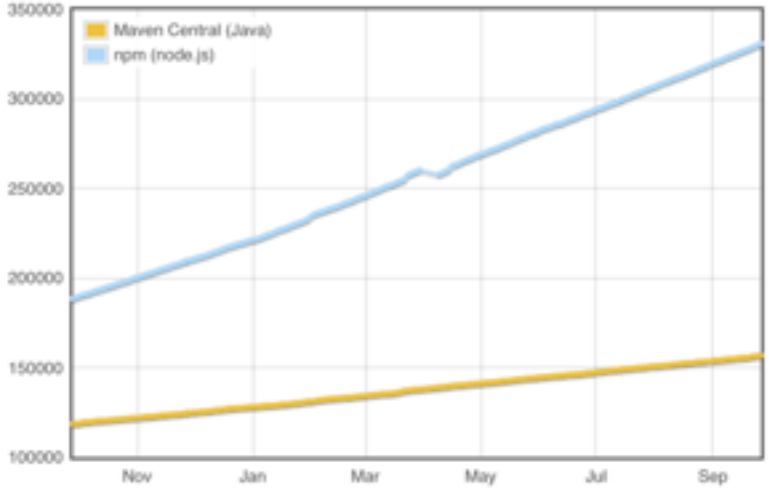
GitHub



#1 JavaScript
#2 Java

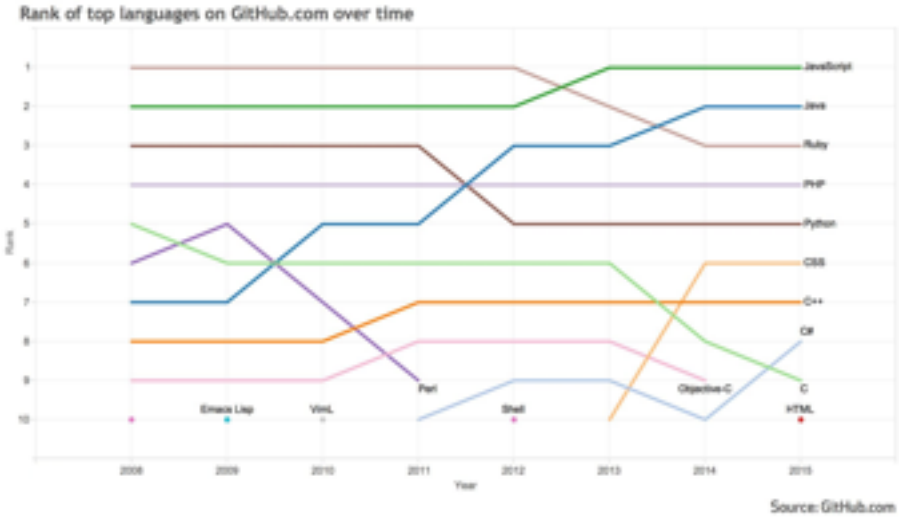
npm vs Maven

Module counts over time



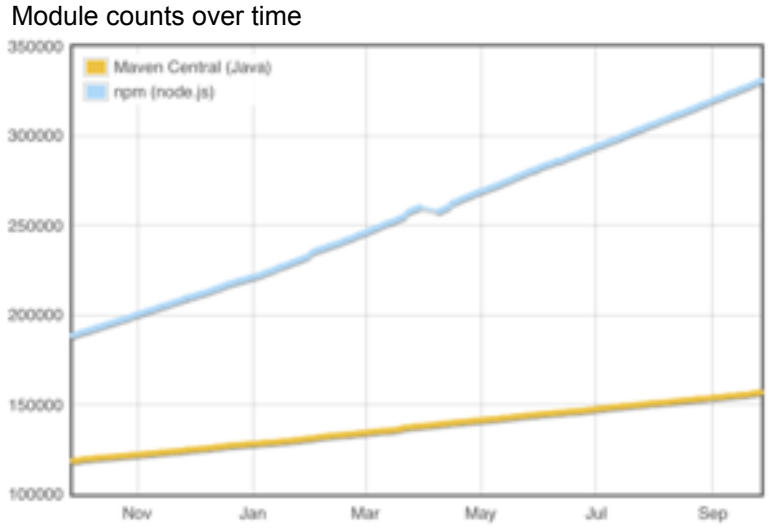
Developer Ecosystem

GitHub



#1 JavaScript
#2 Java

npm vs Maven



#1 JavaScript
#2 Java

Usage in the browser

Unless** it is **absolutely necessary** to run Java **in web browsers, disable it** as described below, even after updating to 7u11. This will help mitigate other Java vulnerabilities that **may be discovered in the future.

*This and previous Java vulnerabilities have been widely targeted by attackers, and new Java vulnerabilities are likely to be discovered. **To defend against** this and **future** Java **vulnerabilities**, **consider disabling Java in web browsers**…*

Usage in the browser

CERT | Software Engineering Institute | Carnegie Mellon University

Homeland Security

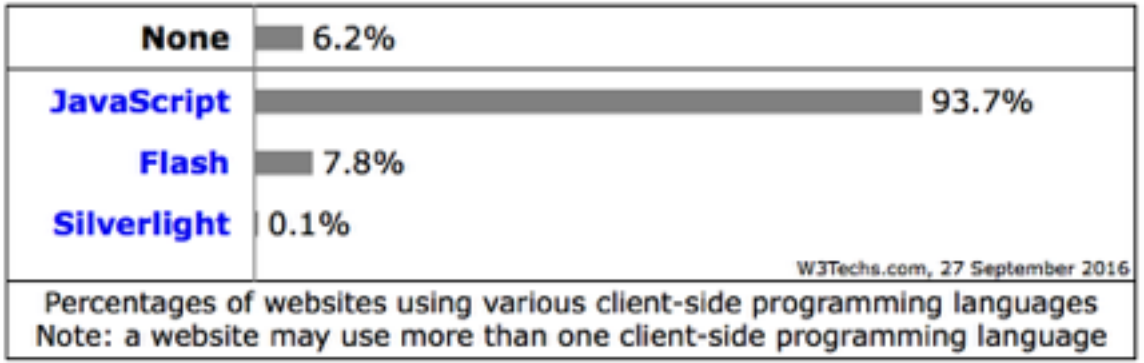
Sponsored by the DHS Office of Cybersecurity and Communications

US-CERT

UNITED STATES COMPUTER EMERGENCY READINESS TEAM

Unless** it is **absolutely necessary** to run Java **in web browsers, disable it** as described below, even after updating to 7u11. This will help mitigate other Java vulnerabilities that **may be discovered in the future.

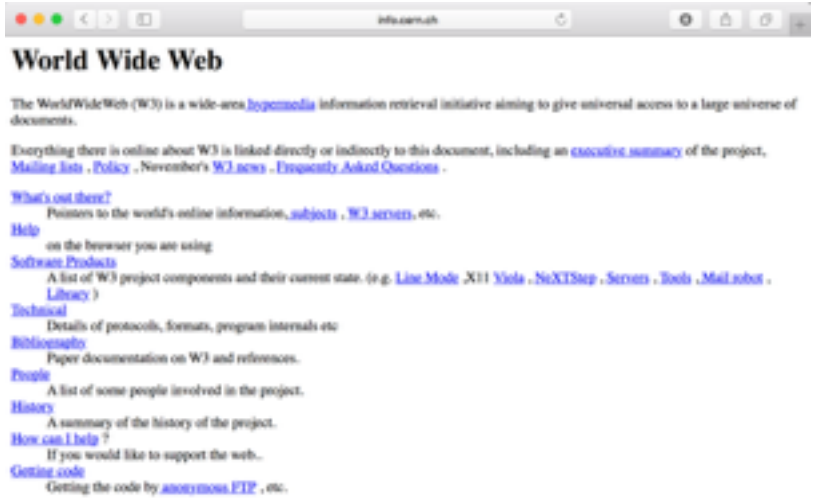
*This and previous Java vulnerabilities have been widely targeted by attackers, and new Java vulnerabilities are likely to be discovered. **To defend against** this and **future** Java **vulnerabilities**, **consider disabling Java in web browsers**…*



Percentages of websites using various client-side programming languages
Note: a website may use more than one client-side programming language

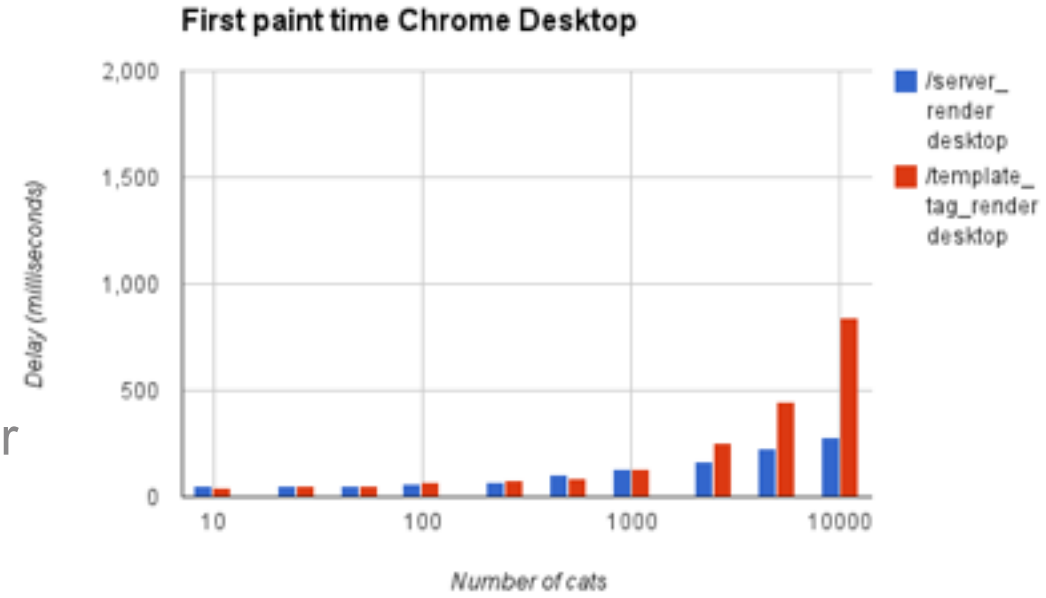
Why JavaScript and Node.js?

Rise of Rich Web Applications



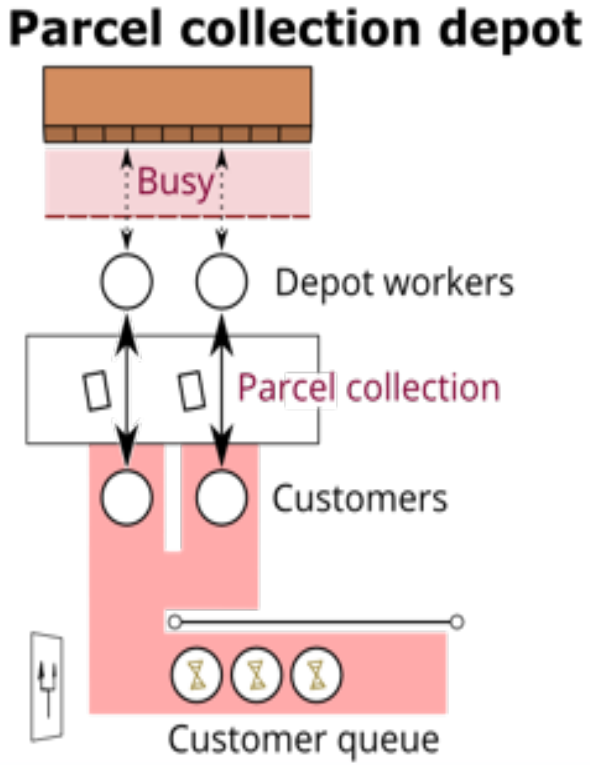
Isomorphic Development an Server-Side Rendering

- Reuse of code components
  - Write One Run Anywhere
- Sharing of data models
  - Less maintenance
- Server-side rendering
  - Pre-Initialise UI on the server
  - Improves first paint time
  - Enables search indexing



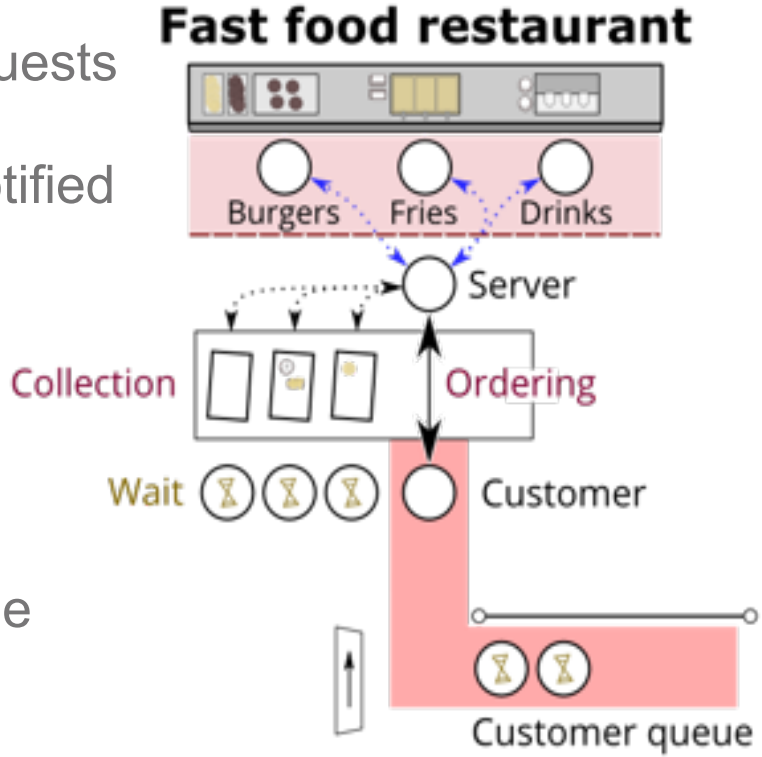
Typical Java Approach to Scalable I/O

- One thread (or process) per connection
  - Each thread waits on a response
  - Scalability determined by the number of threads
- Each thread:
  - consumes memory
  - is relatively idle
- Concurrency determined by number of depot workers



Node.js approach to Scalable I/O

- One thread multiplexes for multiple requests
  - No waiting for a response
  - Handles return from I/O when notified
- Scalability determined by:
  - CPU usage
  - “Back end” responsiveness
- Concurrency determined by how fast the food server can work



Rapid Development

```
var cluster = require('cluster');
var cpus = require('os').cpus().length;
var http = require('http');

if (cluster.isMaster) {
    for (var i = 0; i < cpus; i++) {
        cluster.fork();
    }
    cluster.on('death', function(worker) {
        console.log("Worker" + worker.pid + "died");
    });
} else {
    http.createServer(function(request, response) {
        response.writeHead(200, {"Content-Type": "text/plain"});
        response.write("Hello World!\n");
        response.end();
    }).listen(8080);
}
```

Rapid Development

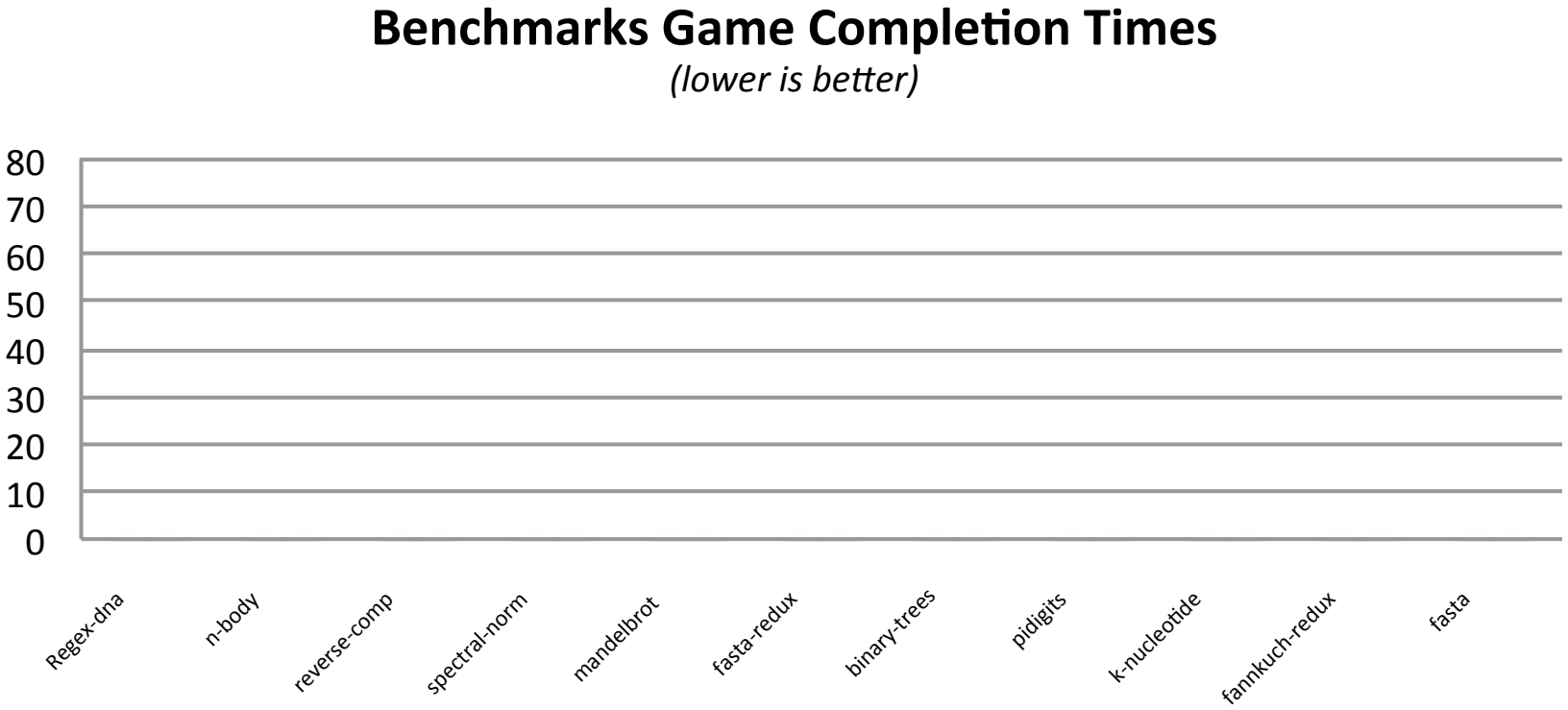
```
var cluster = require('cluster');
var cpus = require('os').cpus().length;
var http = require('http');

if (cluster.isMaster) {
    for (var i = 0; i < cpus; i++) {
        cluster.fork();
    }
    cluster.on('death', function(worker) {
        console.log("Worker" + worker.pid + "died");
    });
} else {
   http.createServer(function(request, response) {
      response.writeHead(200, {"Content-Type": "text/plain"});
      response.write("Hello World!\n");
      response.end();
   }).listen(8080);
}
```

Event based programming

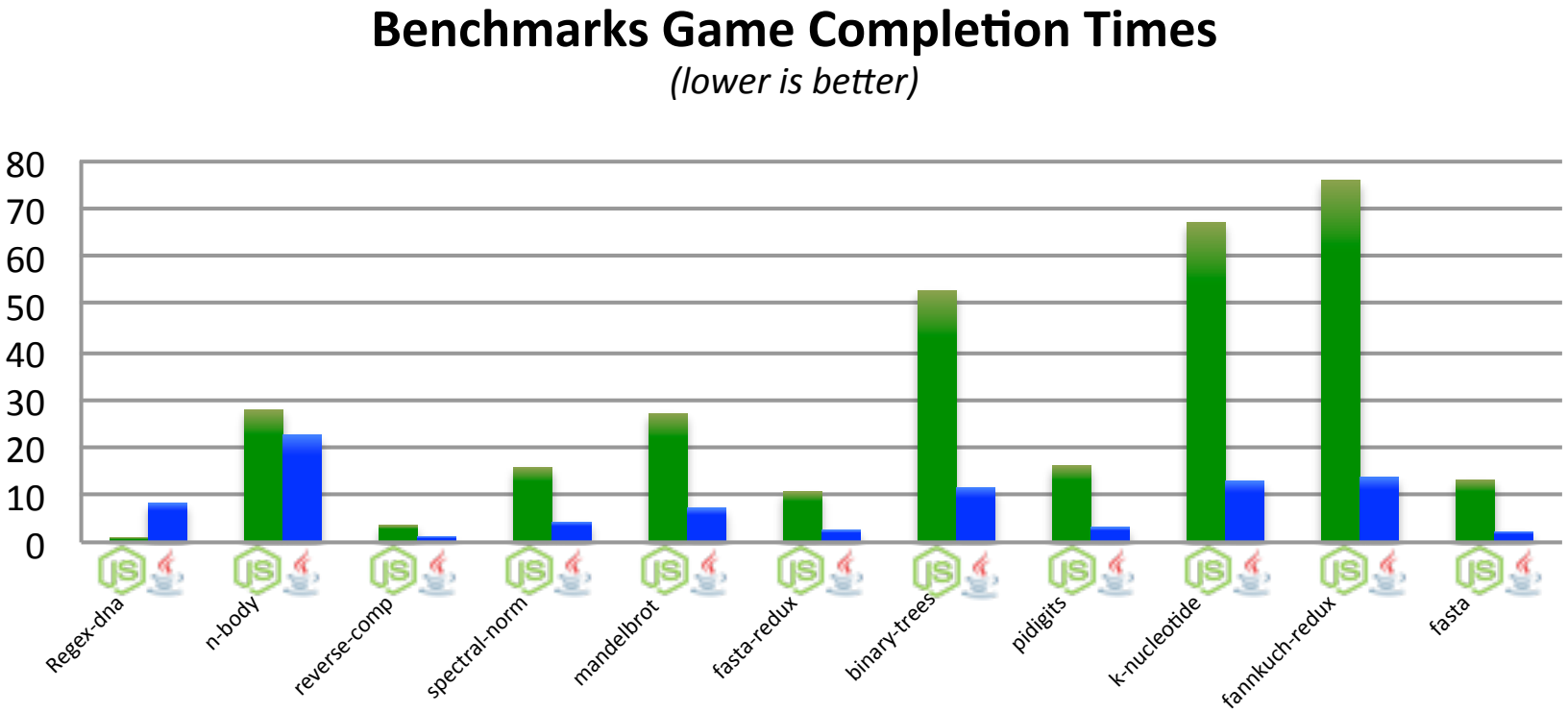
```
var http = require('http');

var server = http.createServer();
server.listen(8080);

server.on('request', function(request, response) {
      response.writeHead(200, {"Content-Type": "text/plain"});
      response.write("Hello World!\n");
      response.end();
});

server.on('connection', function(socket) {});
server.on('close', function() {});
server.on('connect', function(socket) {});
server.on('upgrade', function(request, socket, head) {});
server.on('clientError', function(exception, socket) {});
```

Why Java?

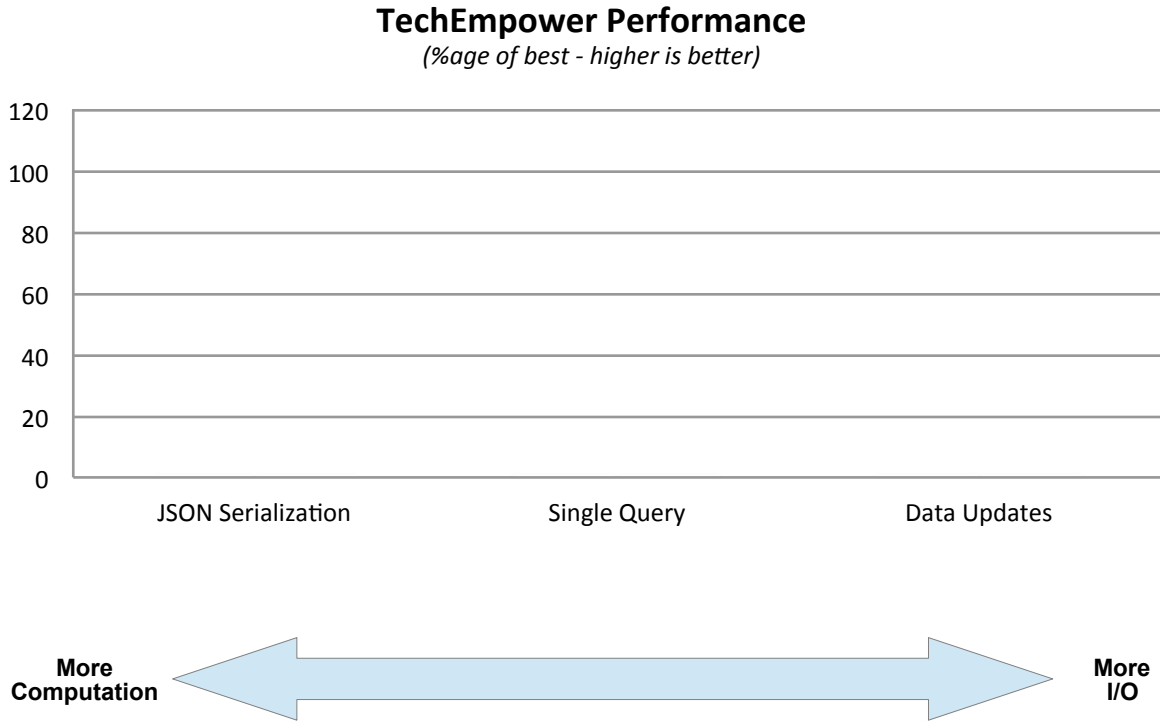
Compute Performance



Compute Performance



Web Application Performance



JSON Serialisation

- JSON serialization of a newly instantiated object
- Maps
  - Key of *message*
  - Value of *Hello, World!*

- Example response:

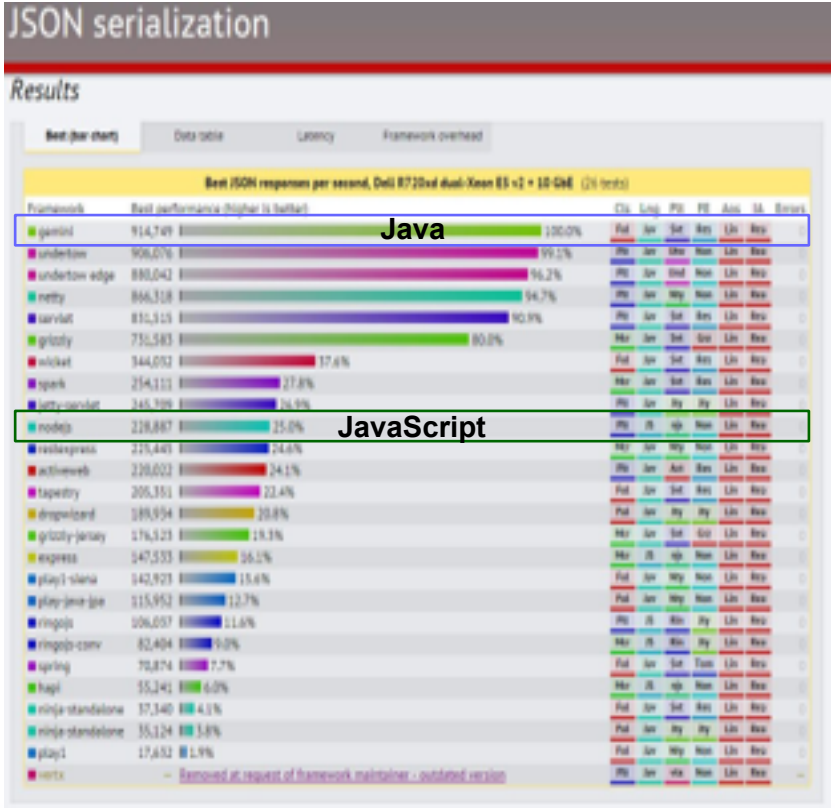
```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 28
Server: Example
Date: Wed, 17 Apr 2013 12:00:00 GMT

{"message":"Hello, World!"}
```

JSON Serialisation

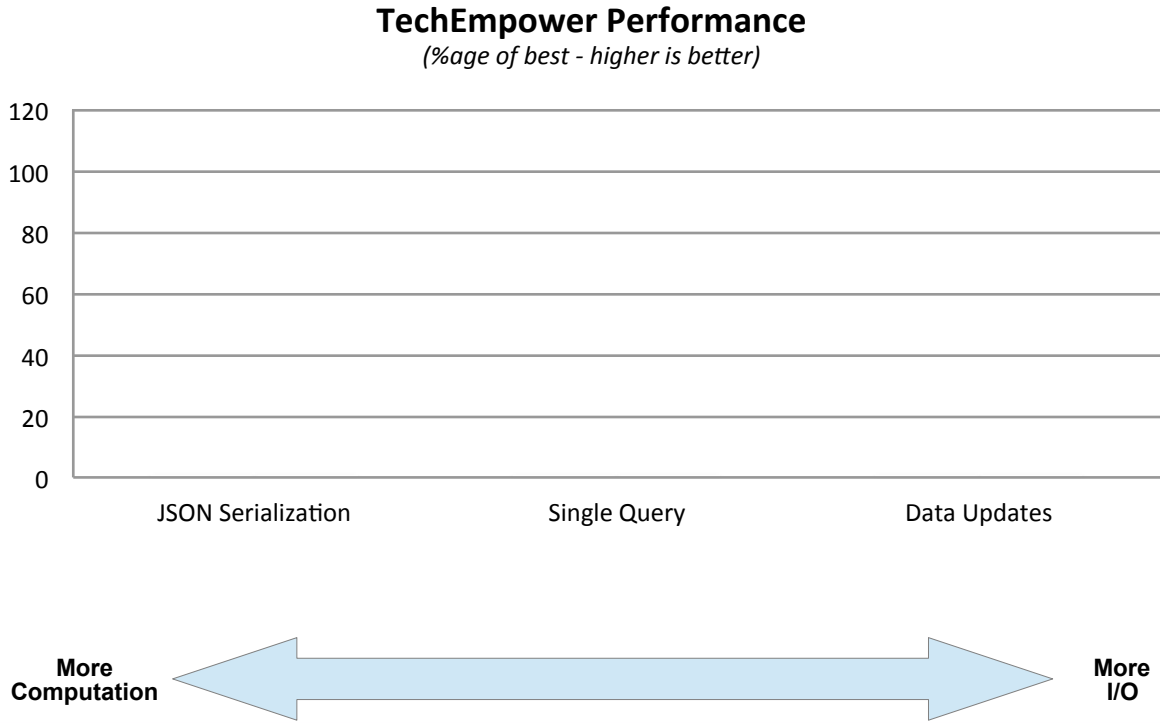
- JSON serialization of a newly instantiated object
- Maps
  - Key of *message*
  - Value of *Hello, World!*

- Example response:

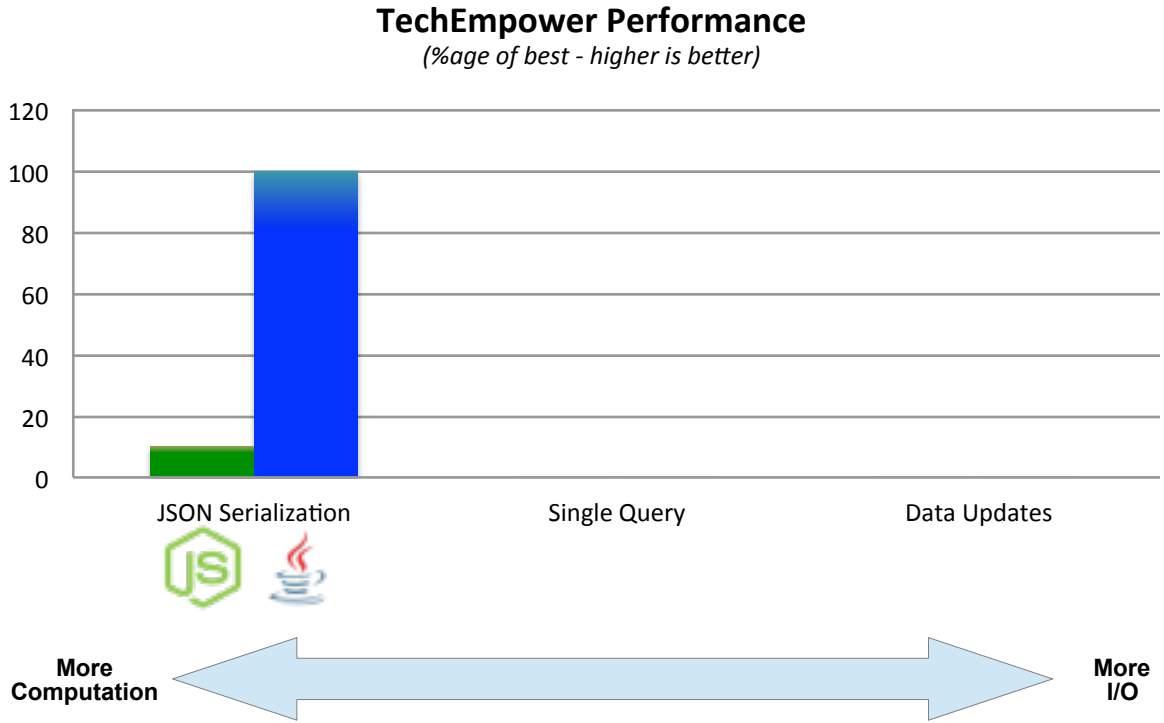
```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 28
Server: Example
Date: Wed, 17 Apr 2013 12:00:00 GMT

{"message":"Hello, World!"}
```



Web Application Performance



Web Application Performance



Single Query

- Fetches single row from simple database table
- Row serialized as JSON

- Example response:

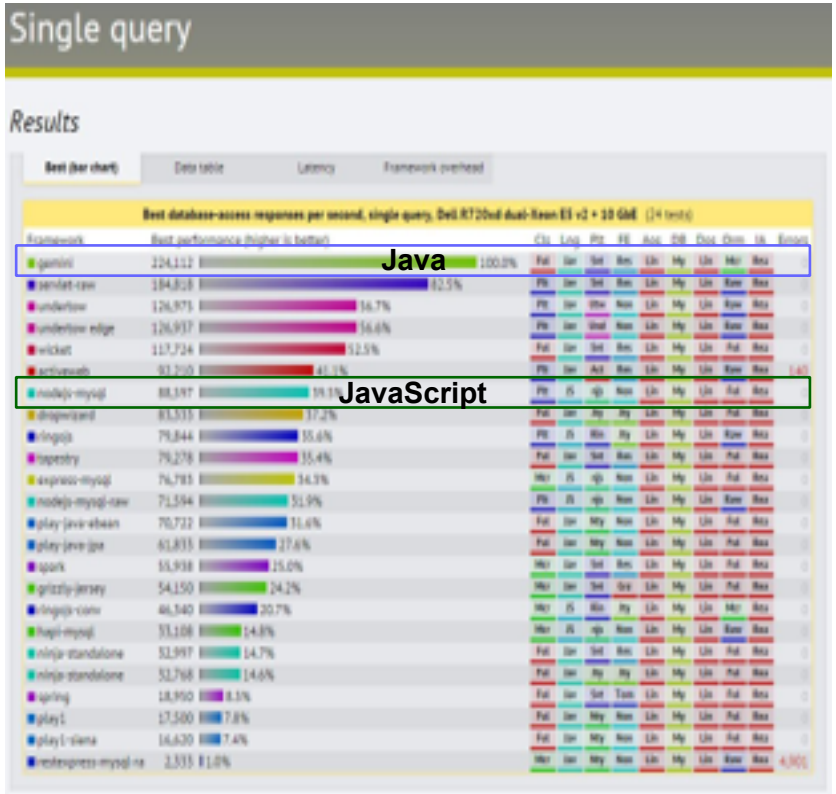
```
HTTP/1.1 200 OK
Content-Length: 32
Content-Type: application/json; charset=UTF-8
Server: Example
Date: Wed, 17 Apr 2013 12:00:00 GMT

{"id":3217,"randomNumber":2149}
```

Single Query

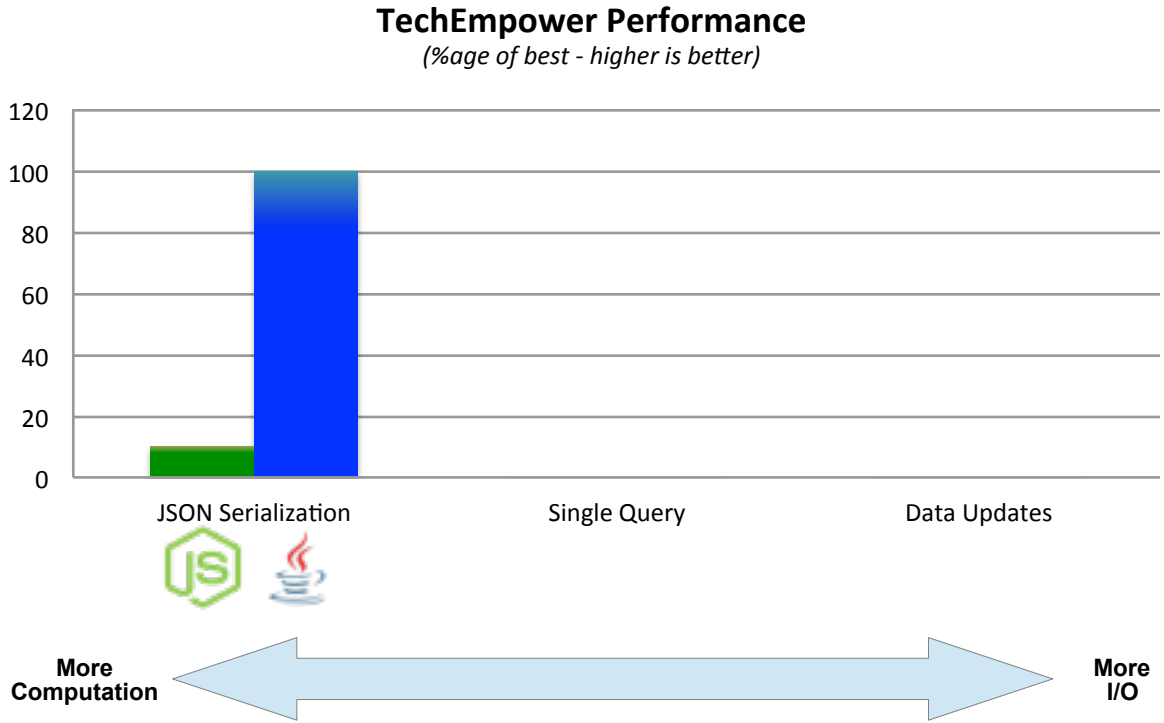
- Fetches single row from simple database table
- Row serialized as JSON

- Example response:

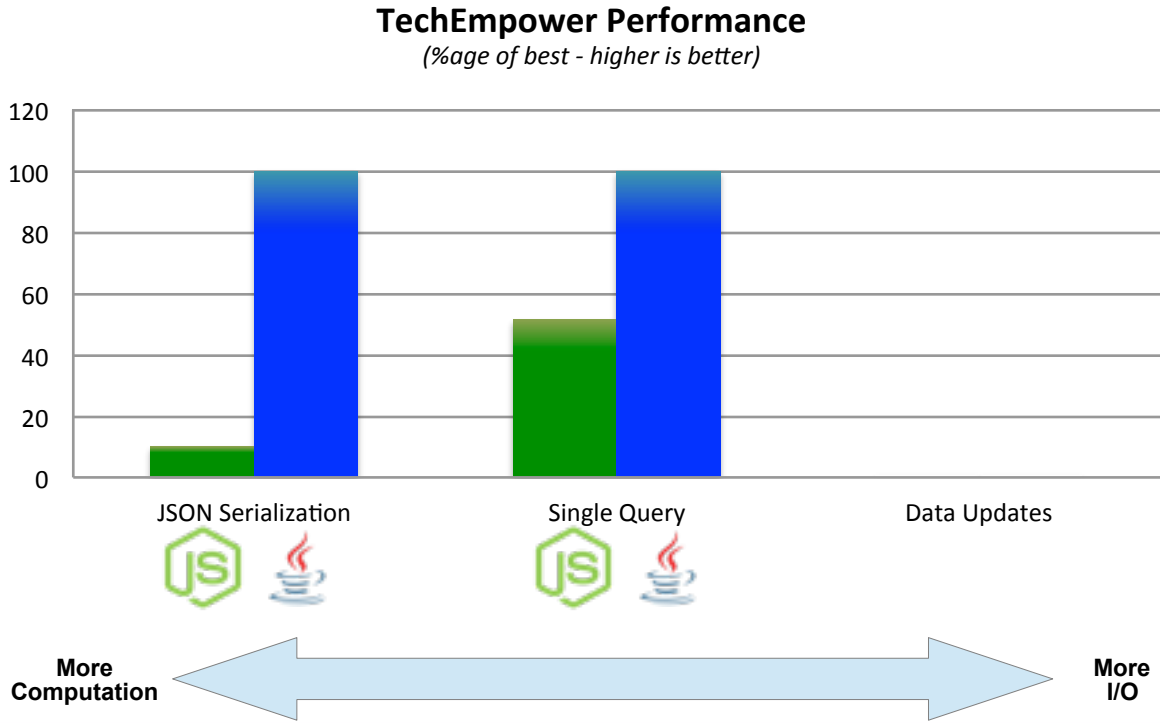
```
HTTP/1.1 200 OK
Content-Length: 32
Content-Type: application/json; charset=UTF-8
Server: Example
Date: Wed, 17 Apr 2013 12:00:00 GMT

{"id":3217,"randomNumber":2149}
```



Web Application Performance

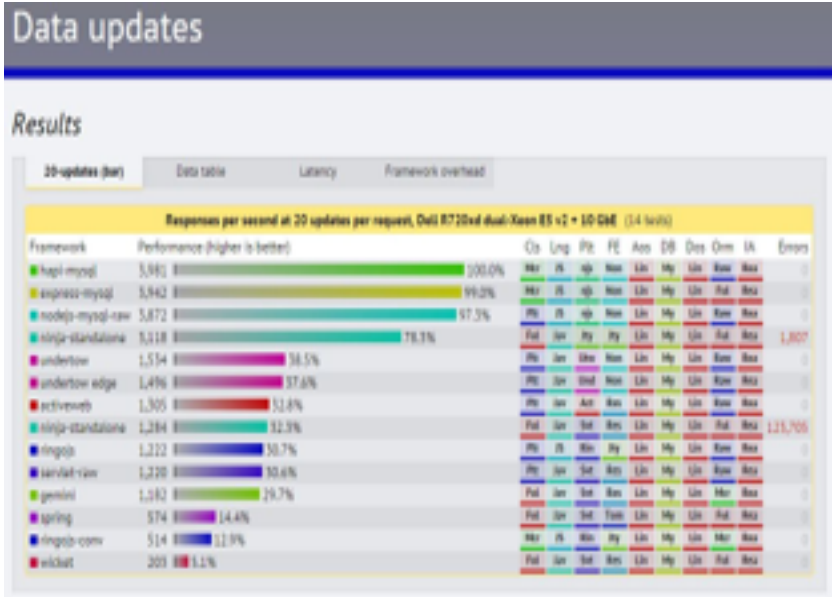


Web Application Performance



Data Updates

- Fetches single row from simple database table
- Row serialized as JSON

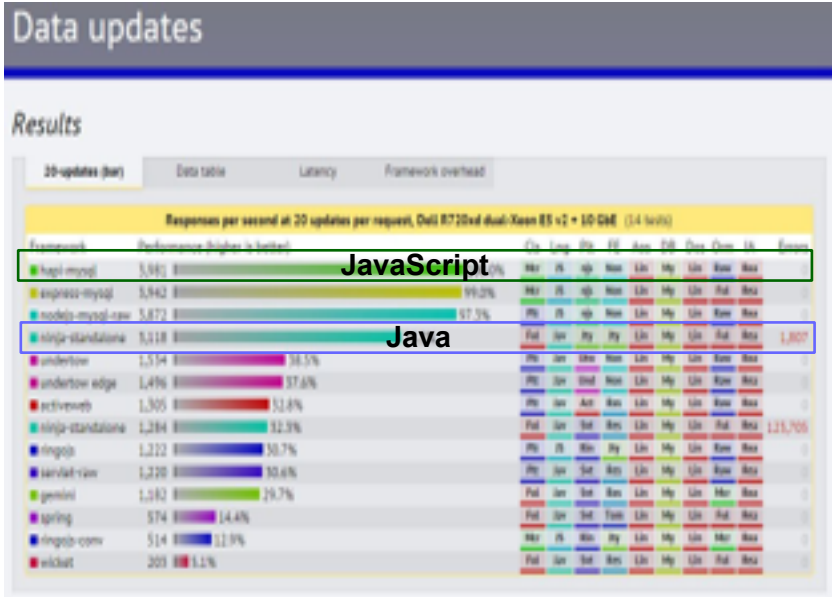


- Example response:

```
HTTP/1.1 200 OK
Content-Length: 315
Content-Type: application/json; charset=UTF-8
Server: Example
Date: Wed, 17 Apr 2013 12:00:00 GMT

[{"id":4174,"randomNumber":331},{"id":51,"randomNumber":6544},{"id":4462,"randomNumber":952},{"id":2131,"randomNumber":532},
{"id":9276,"randomNumber":3097},{"id":3056,"randomNumber":7293},{"id":6964,"randomNumber":620},{"id":675,"randomNumber":6600},
{"id":8414,"randomNumber":6569},{"id":2753,"randomNumber":4065}]
```

Data Updates

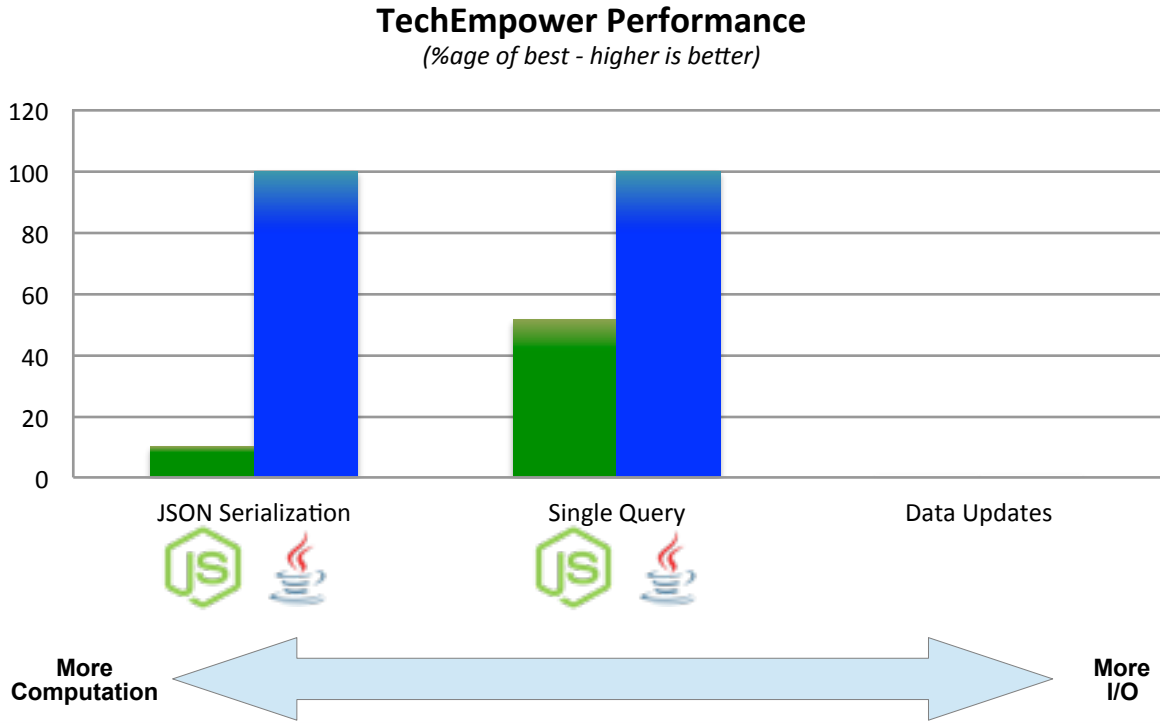
- Fetches single row from simple database table
- Row serialized as JSON



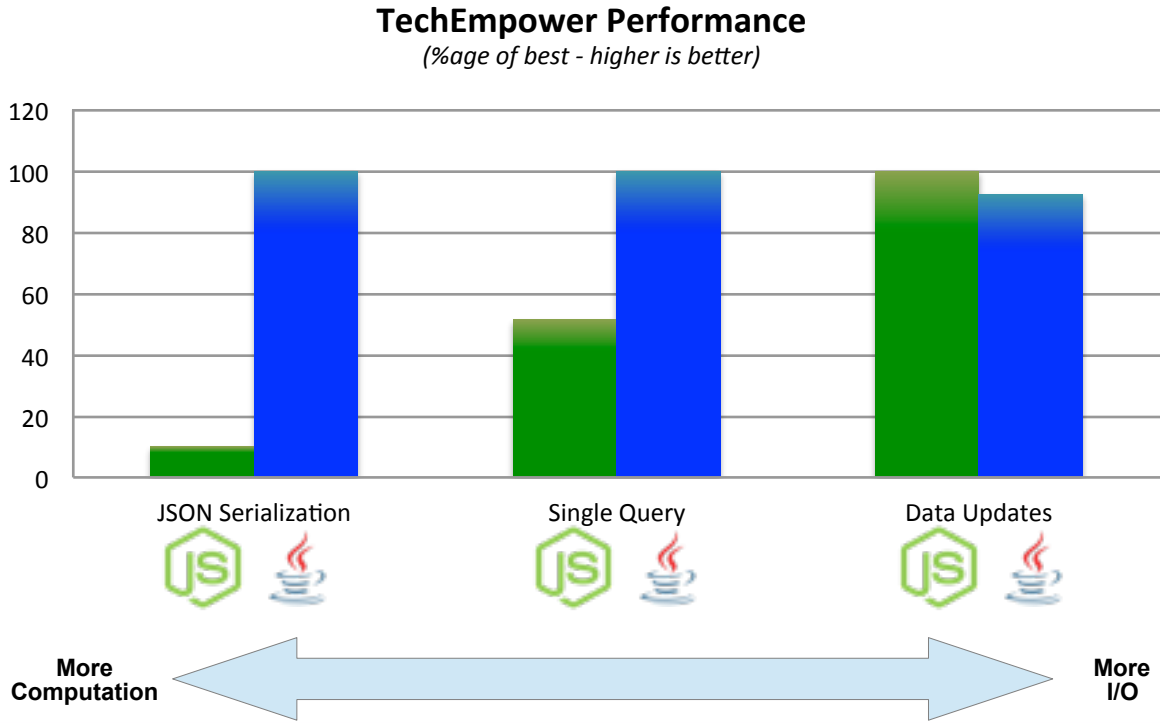
- Example response:

```
HTTP/1.1 200 OK
Content-Length: 315
Content-Type: application/json; charset=UTF-8
Server: Example
Date: Wed, 17 Apr 2013 12:00:00 GMT

[{"id":4174,"randomNumber":331},{"id":51,"randomNumber":6544},{"id":4462,"randomNumber":952},{"id":2221,"randomNumber":532},
{"id":9276,"randomNumber":3097},{"id":3056,"randomNumber":7293},{"id":6964,"randomNumber":620},{"id":675,"randomNumber":6600},
{"id":8414,"randomNumber":6569},{"id":2753,"randomNumber":4065}]
```

Web Application Performance



Web Application Performance

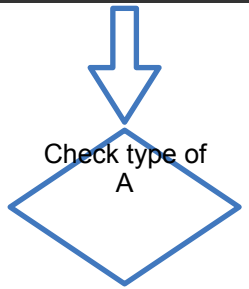


Dynamic Typing in JavaScript: The + operator

```
var add = function (a, b) {
    return (a + b);
}
```

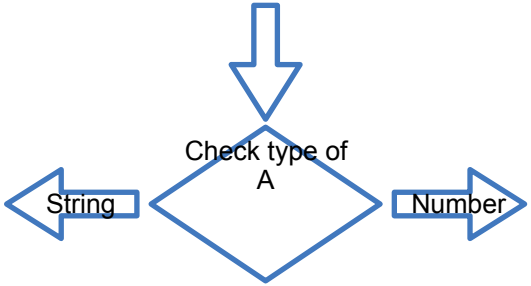
Dynamic Typing in JavaScript: The + operator

```
var add = function (a, b) {
    return (a + b);
}
```



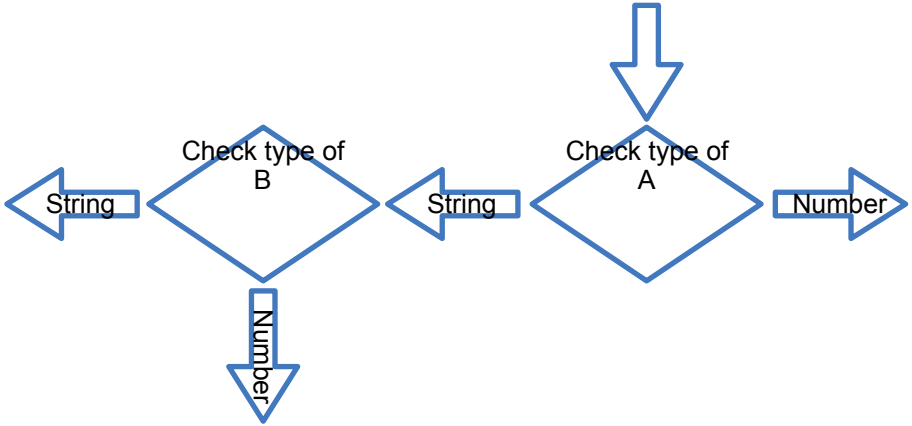
Dynamic Typing in JavaScript: The + operator

```
var add = function (a, b) {
    return (a + b);
}
```



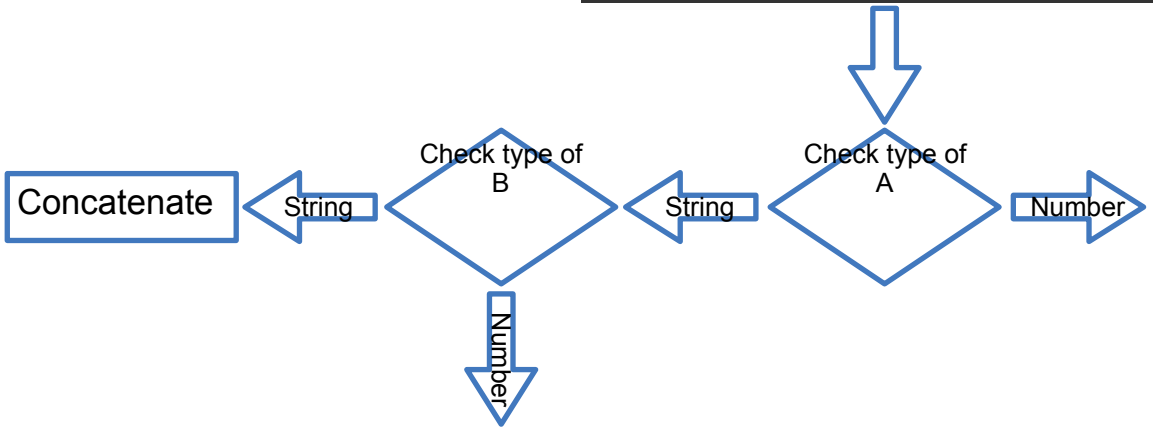
Dynamic Typing in JavaScript: The + operator

```
var add = function (a, b) {
    return (a + b);
}
```



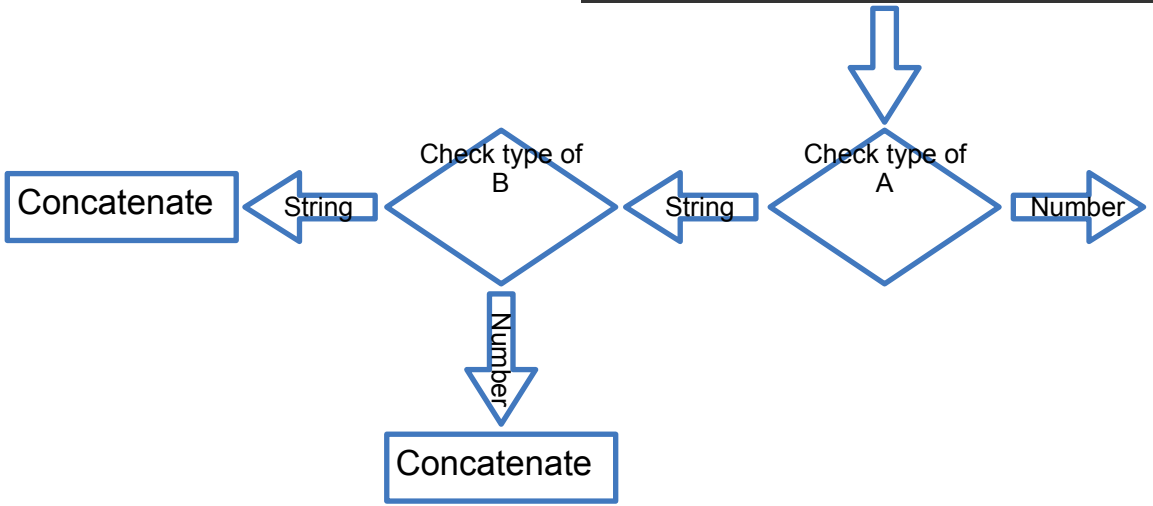
Dynamic Typing in JavaScript: The + operator

```
var add = function (a, b) {
    return (a + b);
}
```



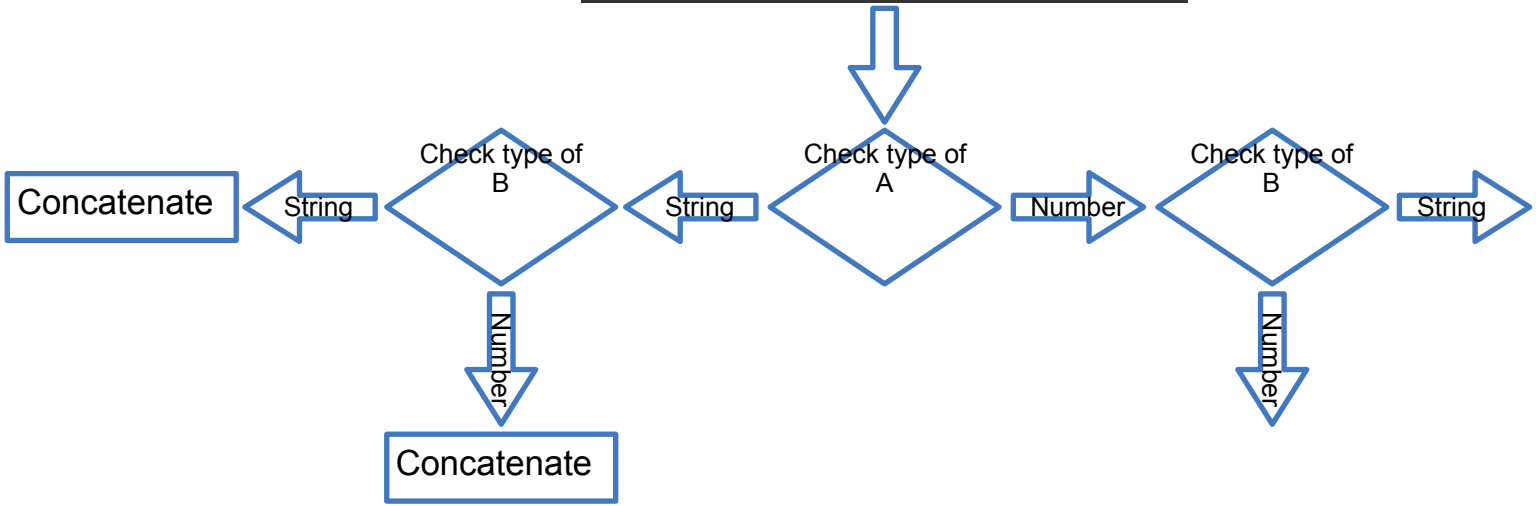
Dynamic Typing in JavaScript: The + operator

```
var add = function (a, b) {
    return (a + b);
}
```



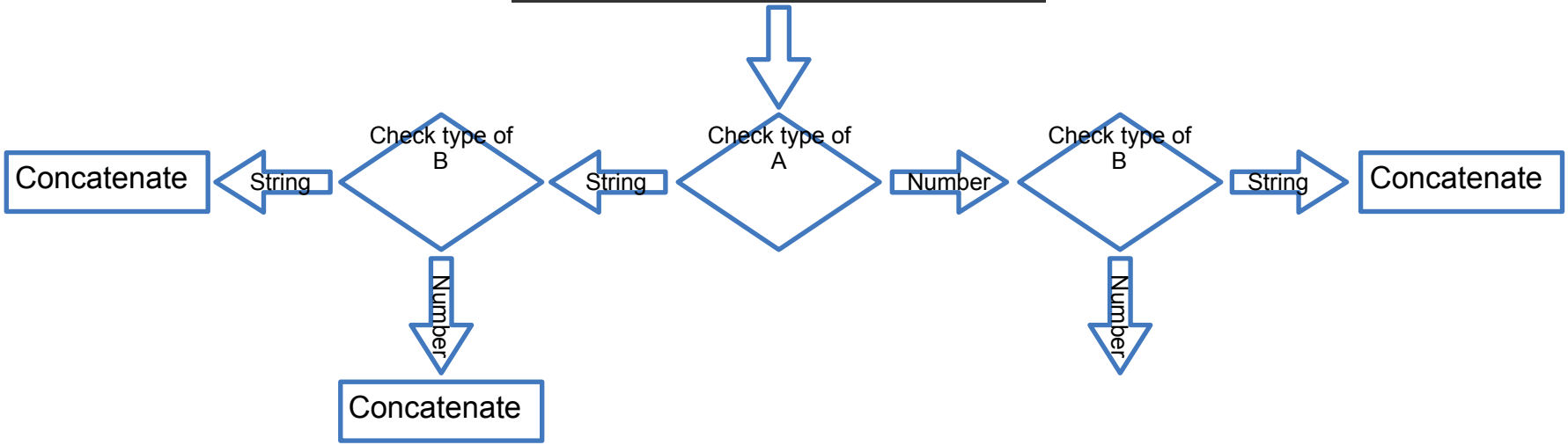
Dynamic Typing in JavaScript: The + operator

```
var add = function (a, b) {
    return (a + b);
}
```



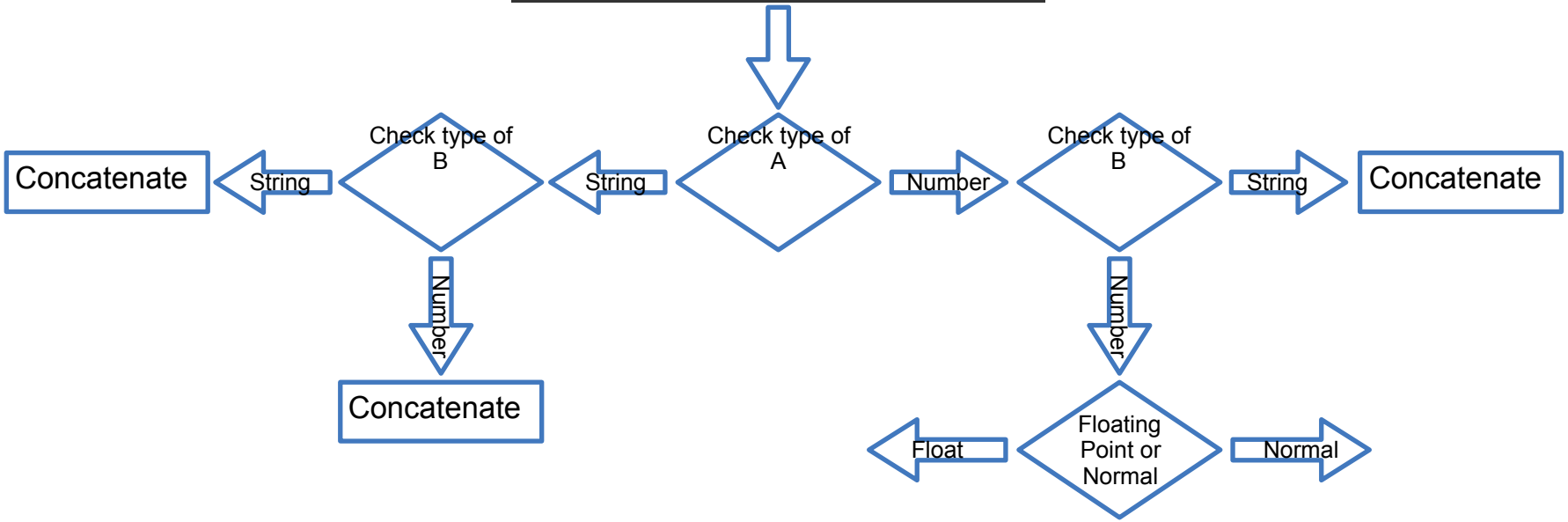
Dynamic Typing in JavaScript: The + operator

```
var add = function (a, b) {
    return (a + b);
}
```



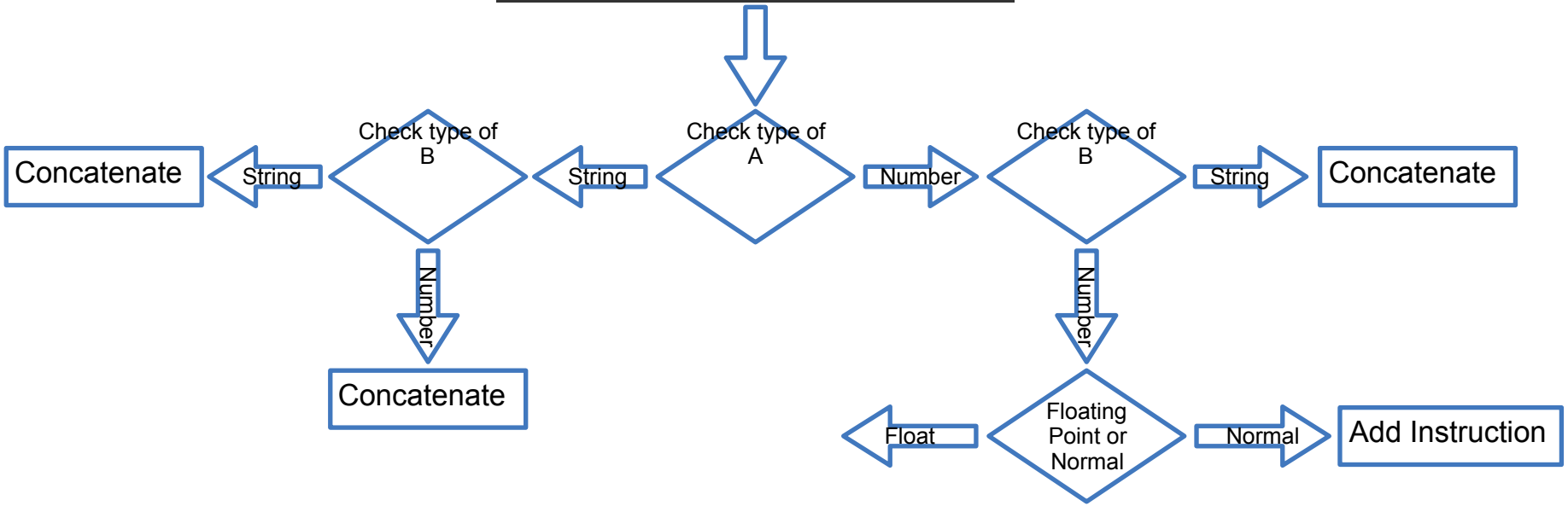
Dynamic Typing in JavaScript: The + operator

```
var add = function (a, b) {
    return (a + b);
}
```



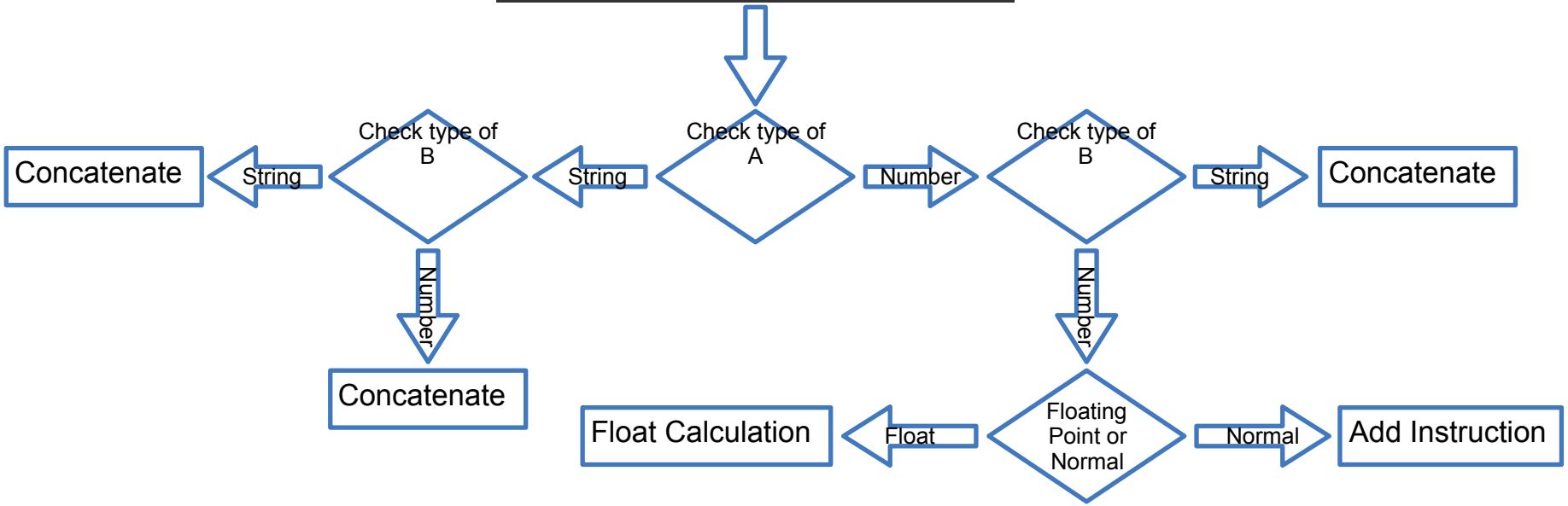
Dynamic Typing in JavaScript: The + operator

```
var add = function (a, b) {
    return (a + b);
}
```



Dynamic Typing in JavaScript: The + operator

```
var add = function (a, b) {
    return (a + b);
}
```

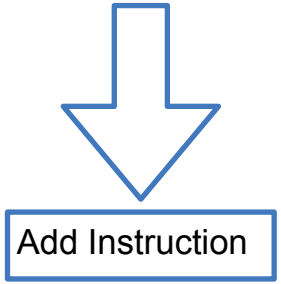


Strong Typing in Java: The + operator

```
int add(int a, int b) {
    return (a + b);
}
```

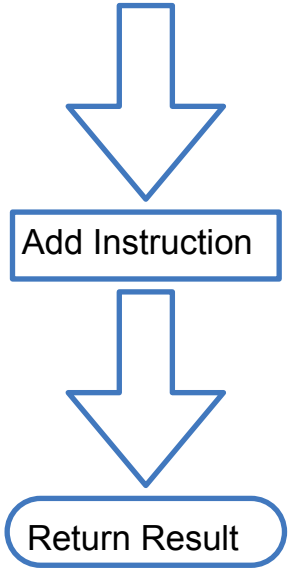
Strong Typing in Java: The + operator

```
int add(int a, int b) {
    return (a + b);
}
```



Strong Typing in Java: The + operator

```
int add(int a, int b) {
    return (a + b);
}
```



Type Safety

```
private static void add (int a, int b){
    System.out.println(a + b);
}
```

```
var add = function (a, b) {
    console.log(a + b);
}
```

Type Safety

```
private static void add (int a, int b){
    System.out.println(a + b);
}

public static void main(String[] args){
    int a = 5;
    int b = 3;

    add(a, b);
}
```

```
var add = function (a, b) {
    console.log(a + b);
}


var a = 5;
var b = 3;

add(a, b);
```

Type Safety

```
private static void add (int a, int b){
    System.out.println(a + b);
}

public static void main(String[] args){
    int a = 5;
    int b = 3;

    add(a, b);
}
```

```
> javac app.java
```

```
var add = function (a, b) {
    console.log(a + b);
}


var a = 5;
var b = 3;

add(a, b);
```

Type Safety

```
private static void add (int a, int b){
    System.out.println(a + b);
}

public static void main(String[] args){
    int a = 5;
    int b = 3;

    add(a, b);
}


> javac app.java
> java app
```

```
var add = function (a, b) {
    console.log(a + b);
}


var a = 5;
var b = 3;

add(a, b);



> node app.js
```

Type Safety

```
private static void add (int a, int b){
    System.out.println(a + b);
}

public static void main(String[] args){
    int a = 5;
    int b = 3;

    add(a, b);
}


> javac app.java
> java app
```

```
> 8
```

```
var add = function (a, b) {
    console.log(a + b);
}


var a = 5;
var b = 3;

add(a, b);




> node app.js
```

```
> 8
```

Type Safety

```
private static void add (int a, int b){
    System.out.println(a + b);
}

public static void main(String[] args){
    String a = new String("5");
    int b = 3;

    add(a, b);
}
```

```
var add = function (a, b) {
    console.log(a + b);
}


var a = '5';
var b = 3;

add(a, b);
```

Type Safety

```
private static void add (int a, int b){
    System.out.println(a + b);
}

public static void main(String[] args){
    String a = new String("5");
    int b = 3;

    add(a, b);
}
```

```
> javac app.java
```

```
var add = function (a, b) {
    console.log(a + b);
}


var a = '5';
var b = 3;

add(a, b);
```

Type Safety

```
private static void add (int a, int b){
    System.out.println(a + b);
}

public static void main(String[] args){
    String a = new String("5");
    int b = 3;

    add(a, b);
}


> javac app.java
Error: incompatible types: String
cannot be converted to int
    add(a, b);
        ^
```

```
var add = function (a, b) {
    console.log(a + b);
}


var a = '5';
var b = 3;

add(a, b);
```

Type Safety

```
private static void add (int a, int b){
    System.out.println(a + b);
}

public static void main(String[] args){
    String a = new String("5");
    int b = 3;

    add(a, b);
}


> javac app.java
Error: incompatible types: String
cannot be converted to int
    add(a, b);
        ^
```

```
var add = function (a, b) {
    console.log(a + b);
}


var a = '5';
var b = 3;

add(a, b);
```

```
> node app.js

> 53
```

JavaScript Calculations

```
> 5 + 3
8

> '5' + 3
'53'
```

JavaScript Calculations

```
> 5 + 3
8

> '5' + 3
'53'

> '5' - 3
```

JavaScript Calculations

```
> 5 + 3
8

> '5' + 3
'53'

> '5' - 3
2                          // String is converted to a number for subtraction
```

JavaScript Calculations

```
> 5 + 3
8

> '5' + 3
'53'

> '5' - 3
2                          // String is converted to a number for subtraction

> '5' - '4'
```

JavaScript Calculations

```
> 5 + 3
8

> '5' + 3
'53'

> '5' - 3
2                          // String is converted to a number for subtraction

> '5' - '4'
1                          // Both Strings converted to number for subtraction
```

JavaScript Calculations

```
> 5 + 3
8

> '5' + 3
'53'

> '5' - 3
2                          // String is converted to a number for subtraction

> '5' - '4'
1                          // Both Strings converted to number for subtraction

> '5' + + '4'
```

JavaScript Calculations

```
> 5 + 3
8

> '5' + 3
'53'

> '5' - 3
2                          // String is converted to a number for subtraction

> '5' - '4'
1                          // Both Strings converted to number for subtraction

> '5' + + '4'
54                         // Multiple +'s are ok
```

JavaScript Calculations

```
> 5 + 3
8

> '5' + 3
'53'

> '5' - 3
2                          // String is converted to a number for subtraction

> '5' - '4'
1                          // Both Strings converted to number for subtraction

> '5' + + '4'
54                         // Multiple +'s are ok

> 'Hello' + 'World'
```

JavaScript Calculations

```
> 5 + 3
8

> '5' + 3
'53'

> '5' - 3
2                          // String is converted to a number for subtraction

> '5' - '4'
1                          // Both Strings converted to number for subtraction

> '5' + + '4'
54                         // Multiple +'s are ok

> 'Hello' + 'World'
'HelloWorld'               // Ok, that's expected
```

JavaScript Calculations

```
> 5 + 3
8

> '5' + 3
'53'

> '5' - 3
2                          // String is converted to a number for subtraction

> '5' - '4'
1                          // Both Strings converted to number for subtraction

> '5' + + '4'
54                         // Multiple +'s are ok

> 'Hello' + 'World'
'HelloWorld'               // Ok, that's expected

> 'Hello' + + 'World'
```

JavaScript Calculations

```
> 5 + 3
8

> '5' + 3
'53'

> '5' - 3
2                          // String is converted to a number for subtraction

> '5' - '4'
1                          // Both Strings converted to number for subtraction

> '5' + + '4'
54                         // Multiple +'s are ok

> 'Hello' + 'World'
'HelloWorld'               // Ok, that's expected

> 'Hello' + + 'World'
'HelloNaN'                 // Multiple plus must cause String to number conversion
```

JavaScript Calculations

```
> '5' + - '5'
'5-2'                          // I can just about see that works
```

JavaScript Calculations

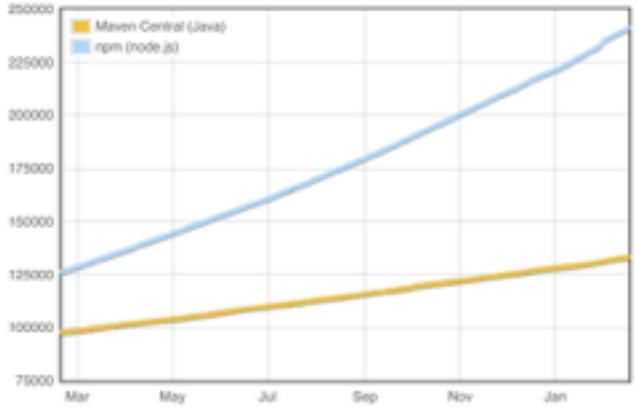
```
> '5' + - '5'
'5-2'                          // I can just about see that works

> var x = 3
> '5' - x + x
5                              // Ok, that makes sense
```

JavaScript Calculations

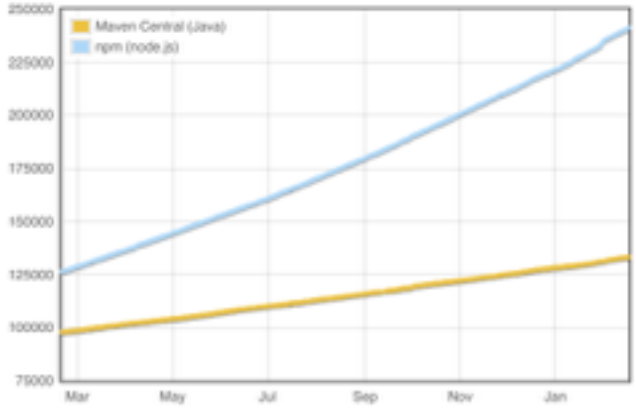
```
> '5' + - '5'
'5-2'                          // I can just about see that works

> var x = 3
> '5' - x + x
5                              // Ok, that makes sense

> var x = 3
> '5' + x - x
50                             // What???
```

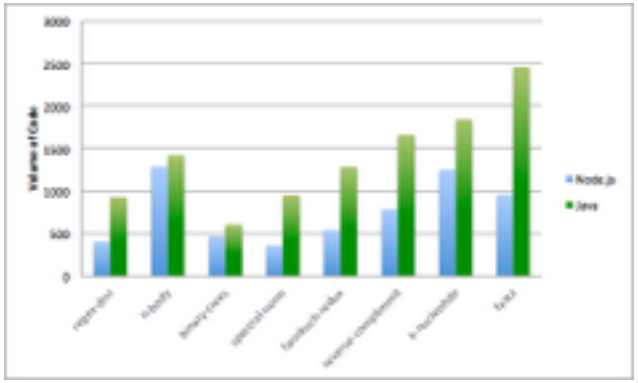
Language Selection

Choosing the Right Language for the Service

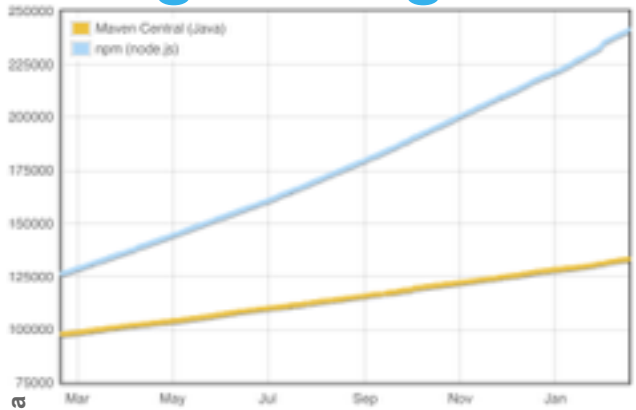


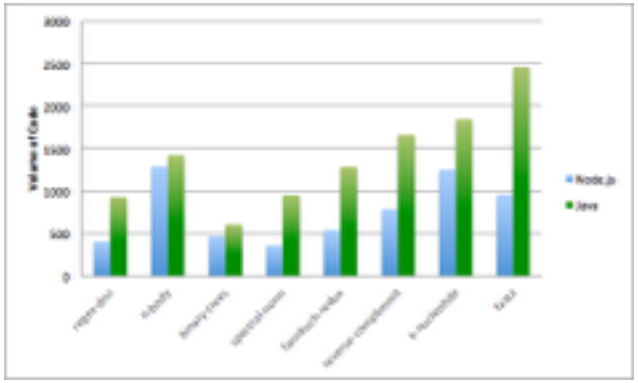
Choosing the Right Language for the Service



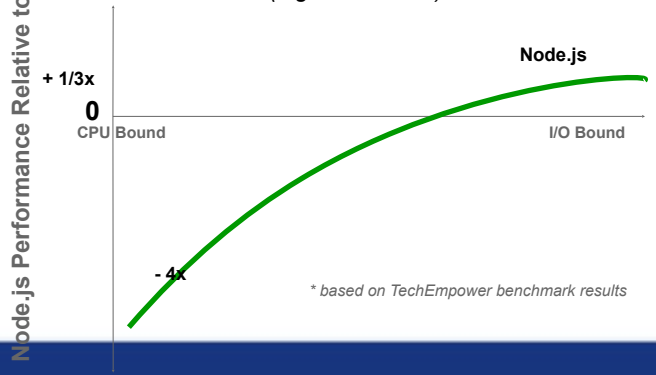


Choosing the Right Language for the Service

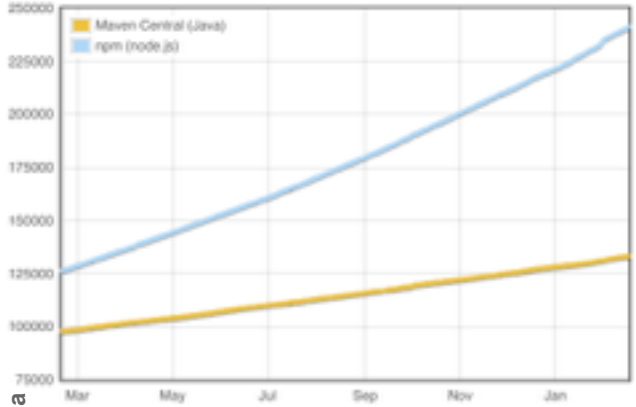


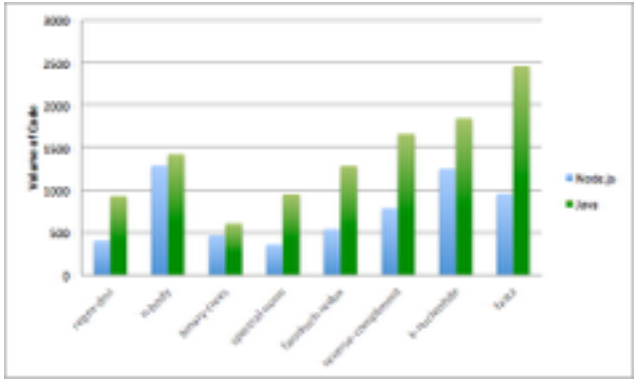


Application Performance
(higher is better)

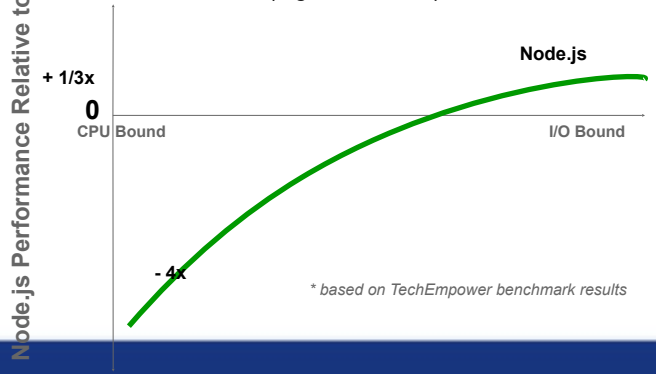


Choosing the Right Language for the Service





Application Performance
(higher is better)




```
Error: incompatible types
ClassCastException
```

Microservices Paradigm

“Do one thing, and do it well”

Choosing the Right Language for the Service

- Higher performance for I/O
- Easier async programming
- Fullstack/isomorphic development

Choosing the Right Language for the Service

- Higher processing performance
- Type safety for calculations
- Rich processing frameworks

Choosing the Right Language for the Service

+

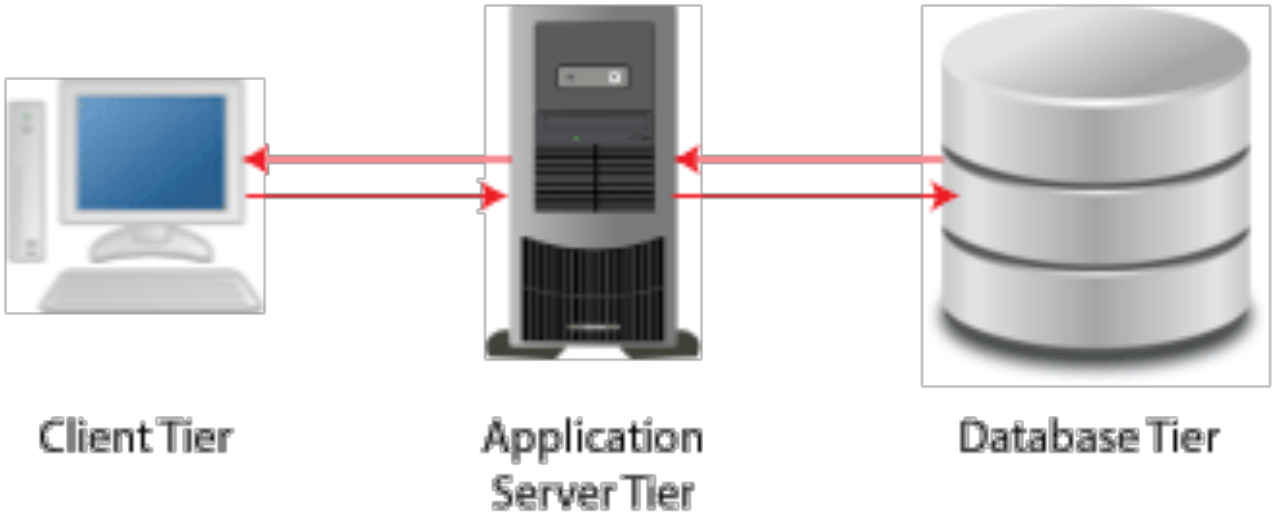
Highly performant, scalable rich web applications

Highly performant, reliable transaction processing

Self-contained micro-service components

Emerging Architectures

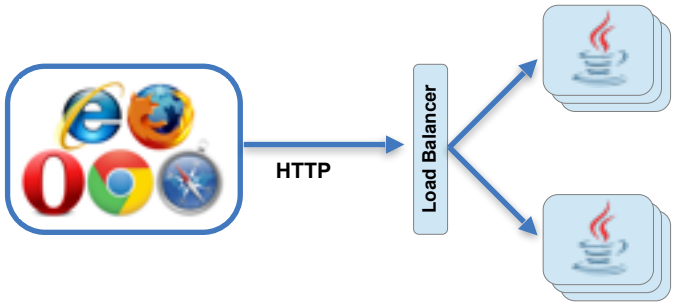
Rich Web Applications



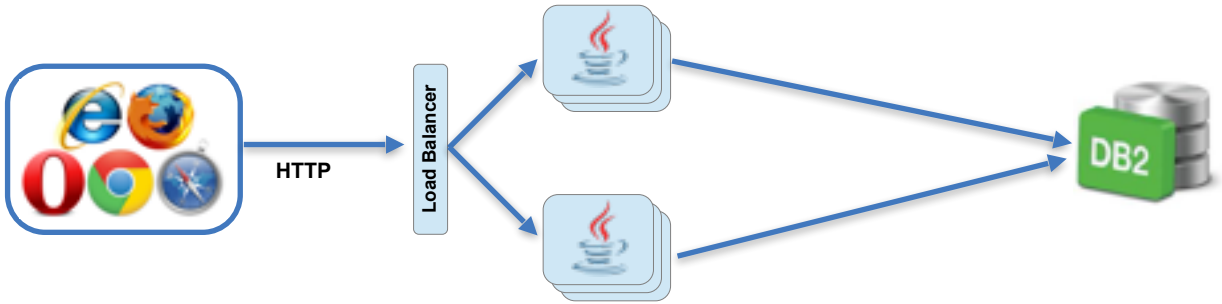
Rich Web Applications

Rich Web Applications

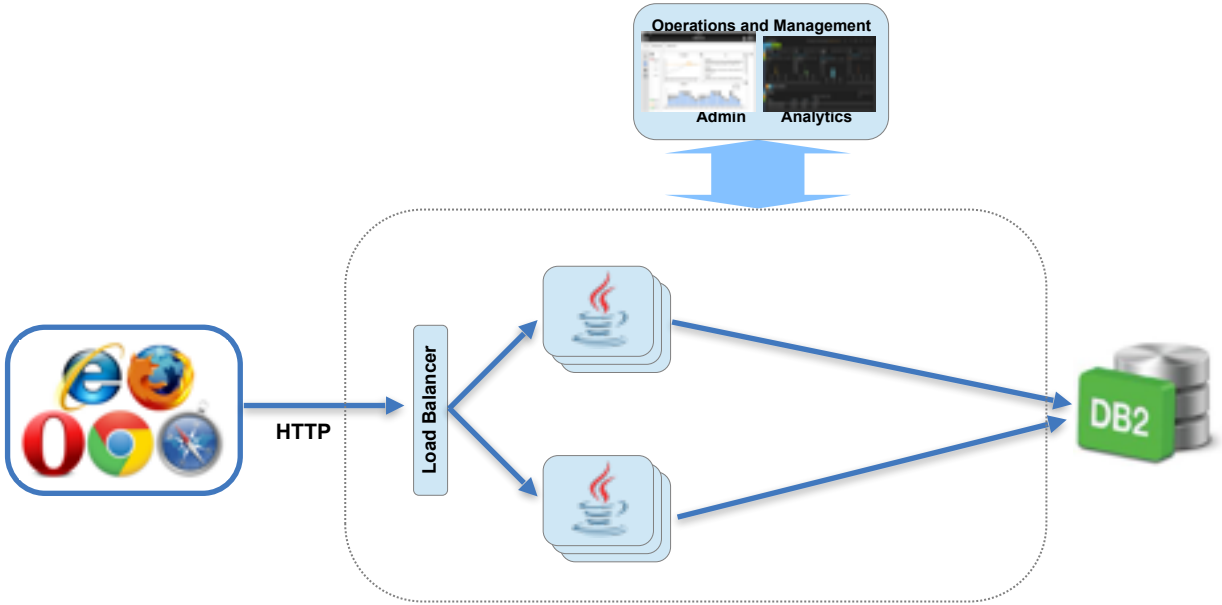
Rich Web Applications



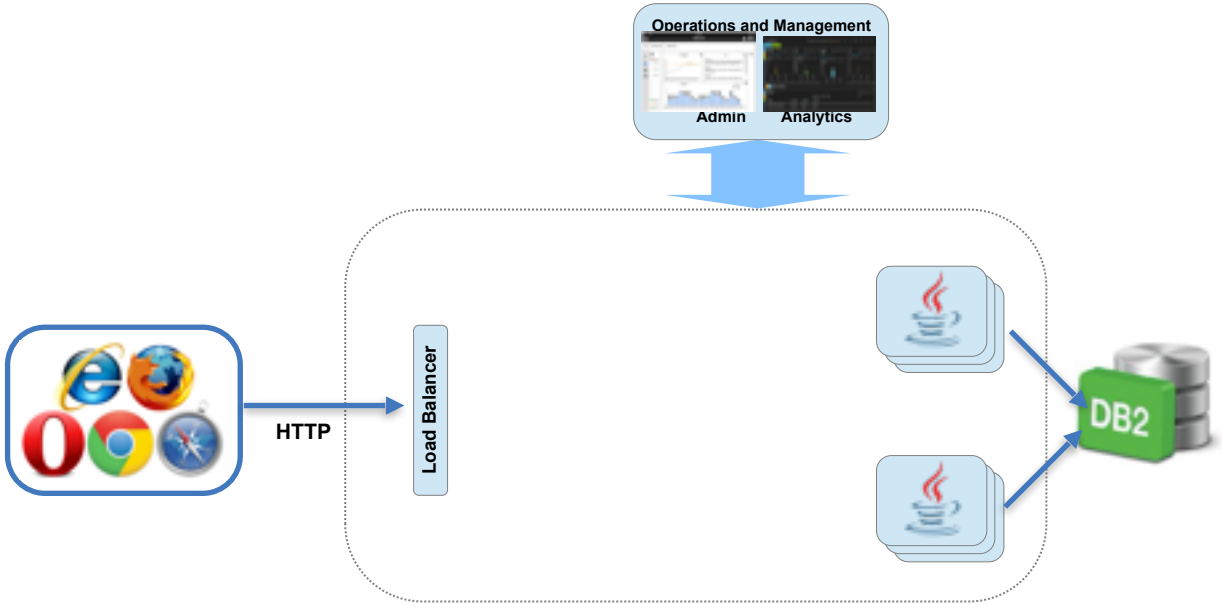
Rich Web Applications



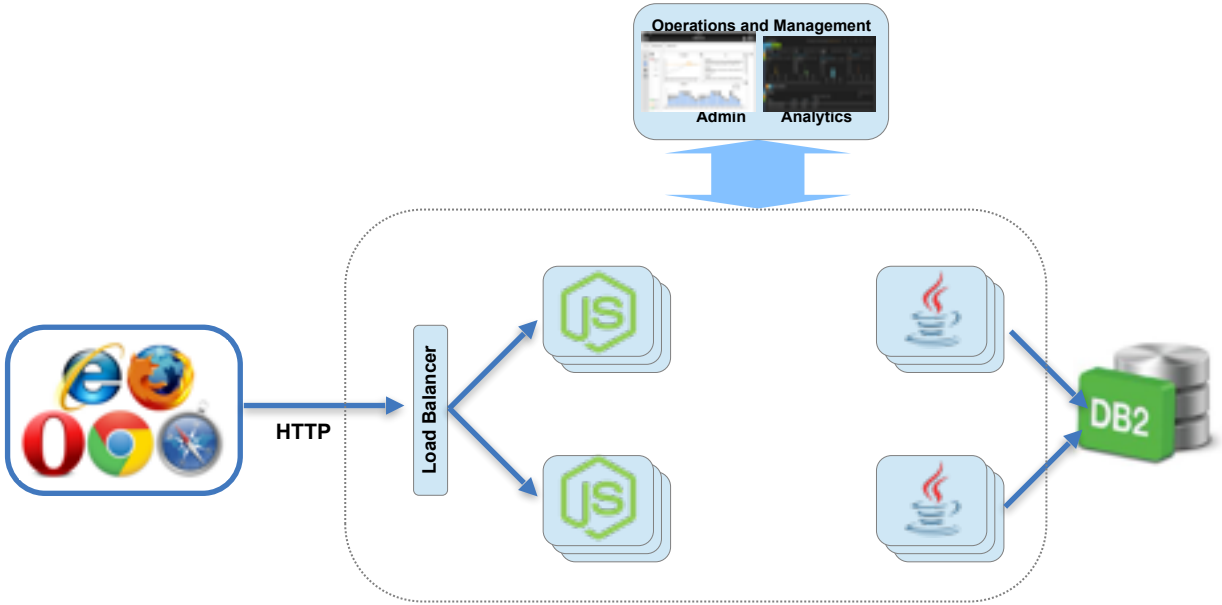
Rich Web Applications



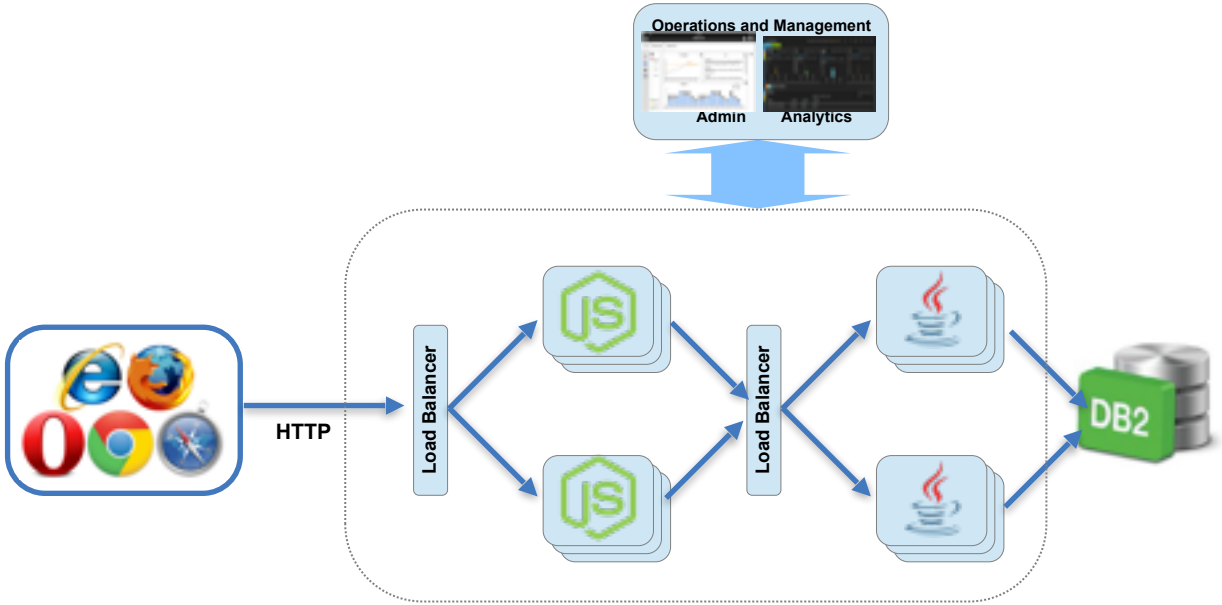
Rich Web Applications



Rich Web Applications



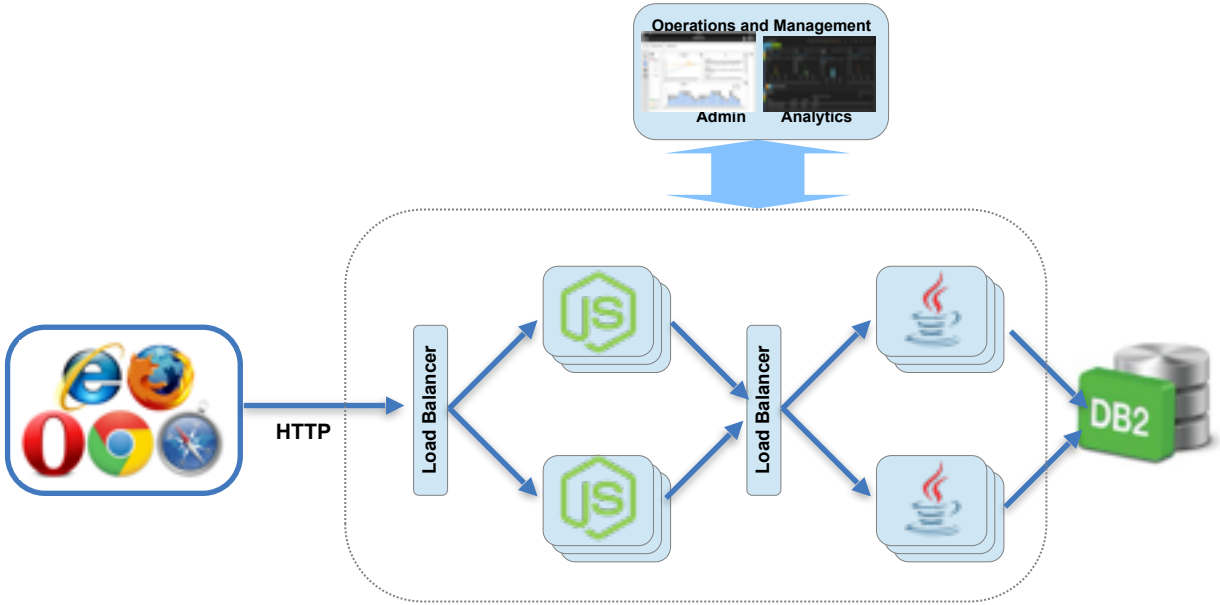
Rich Web Applications



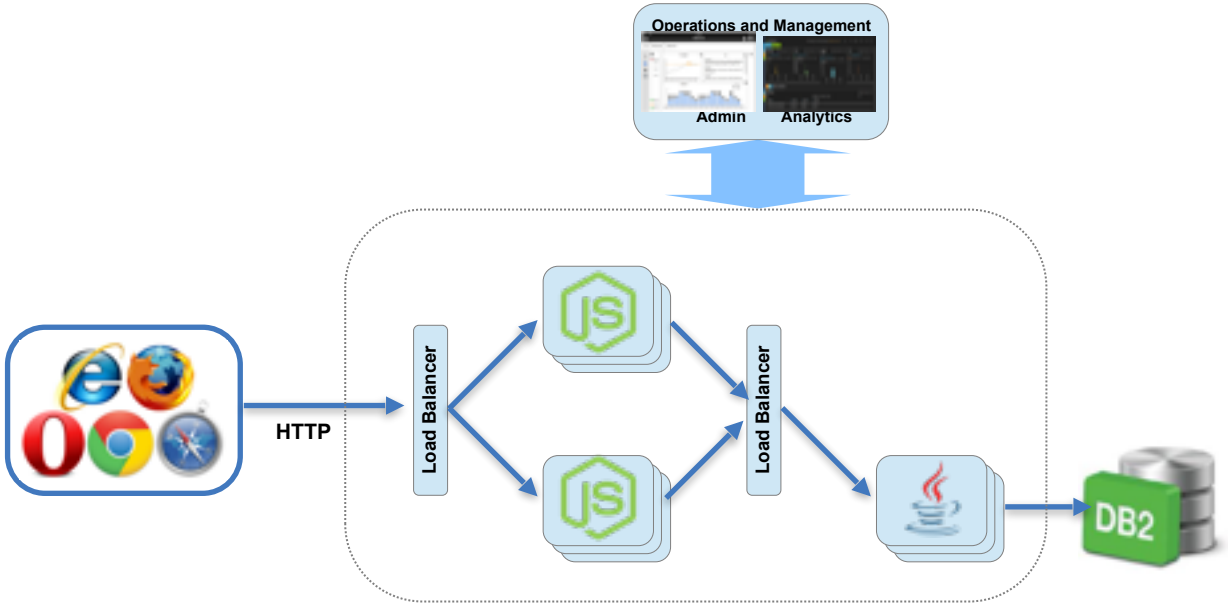
Forrester 4-Tier Architecture

Client tier		• Mobile clients • Wearables • Internet of things • Responsible for experience delivery
Delivery tier		• Optimizes content for proper display on device • Caches content for performant delivery • Drives personalization by using analytics to monitor user behavior
Aggregation tier		• Aggregates and federates services tier data • Provides discovery for the underlying service library • Performs data protocol translation (e.g., SOAP to JSON)
Services tier		• Existing on-premises systems of record, services, and data • External third-party services (e.g., Box, Twilio, Urban Airship)

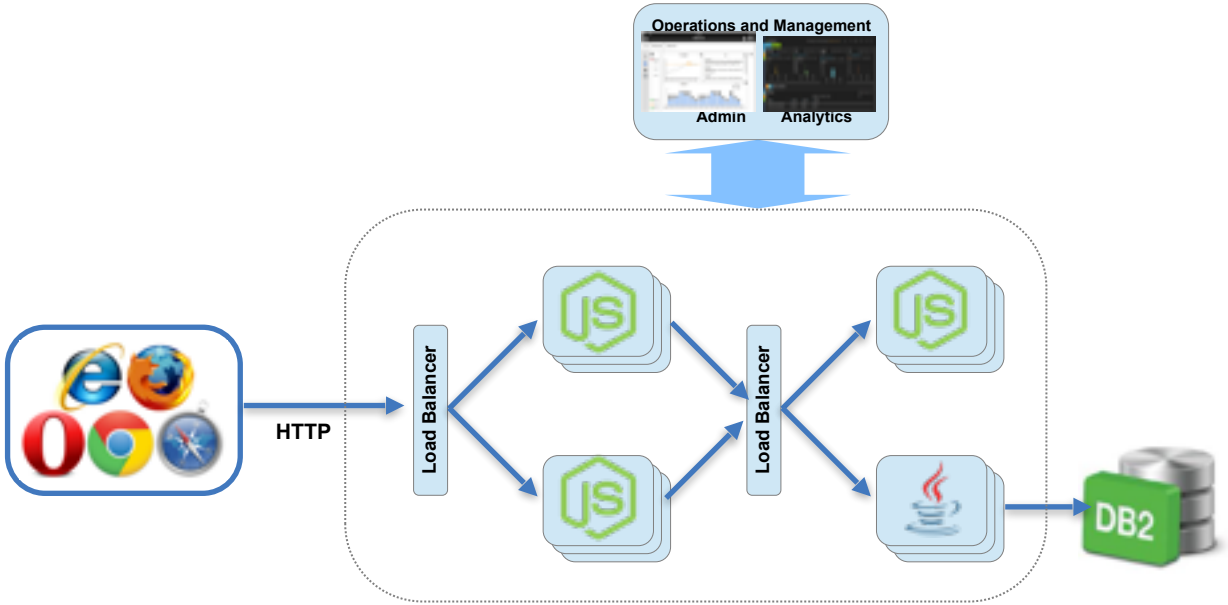
MicroServices and API Economy



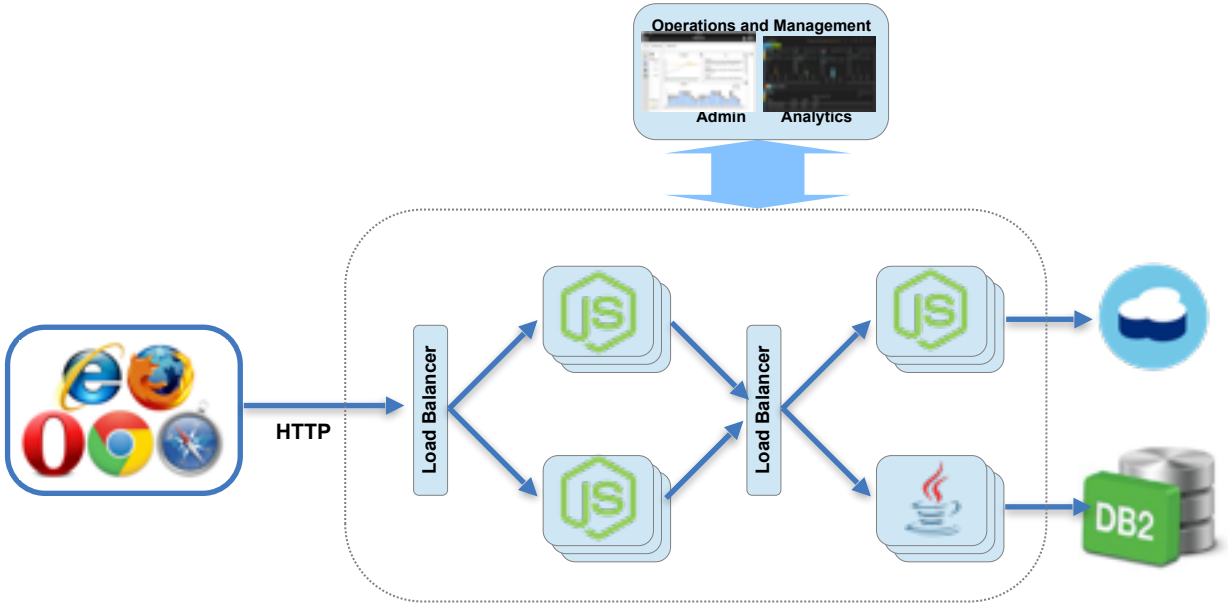
MicroServices and API Economy



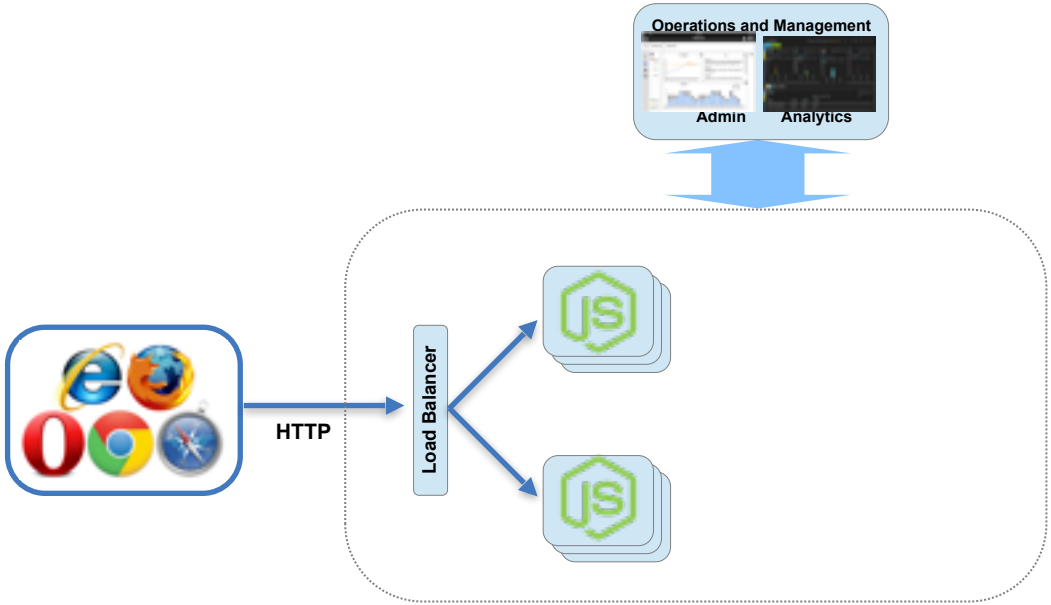
MicroServices and API Economy



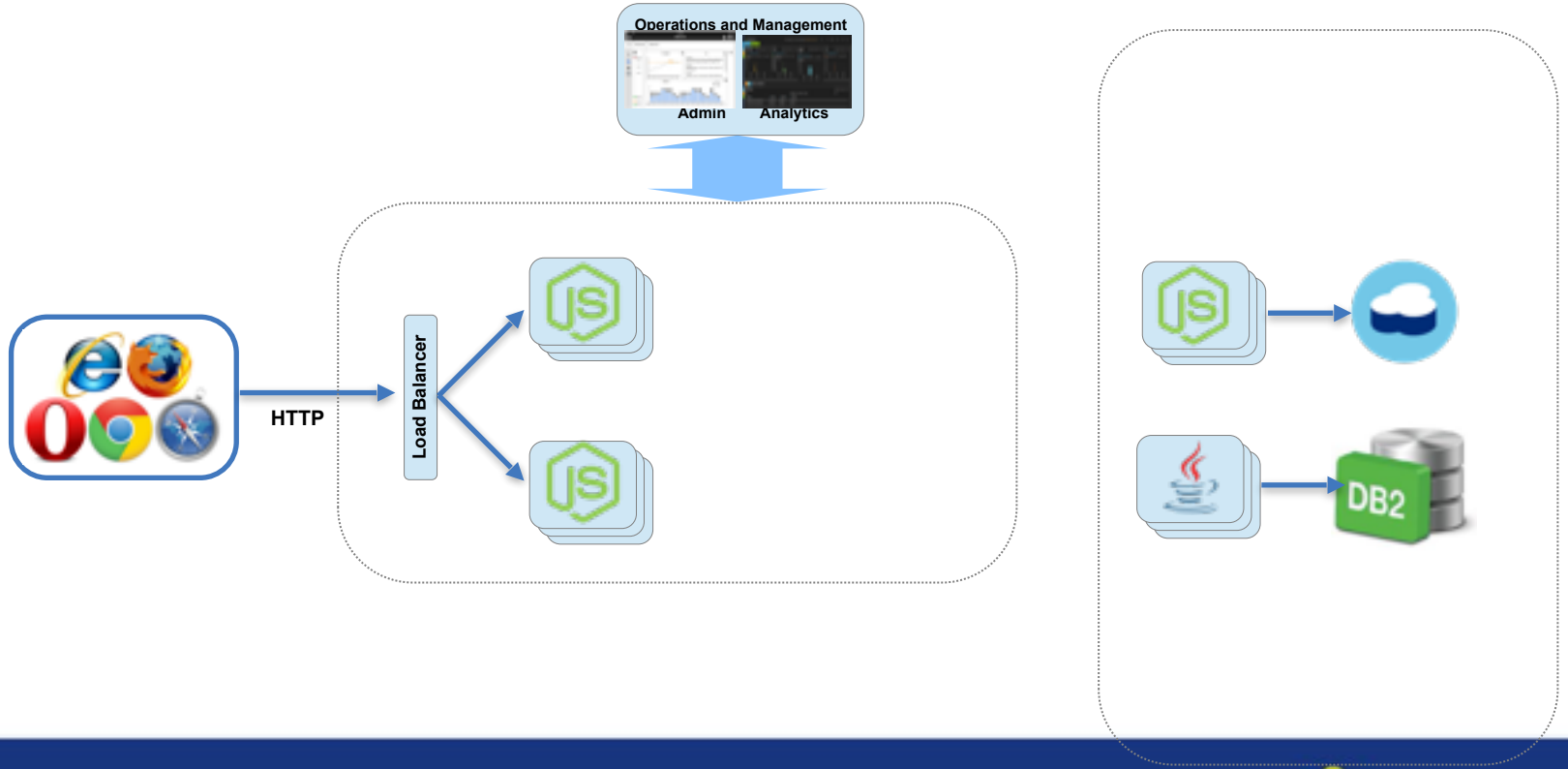
MicroServices and API Economy



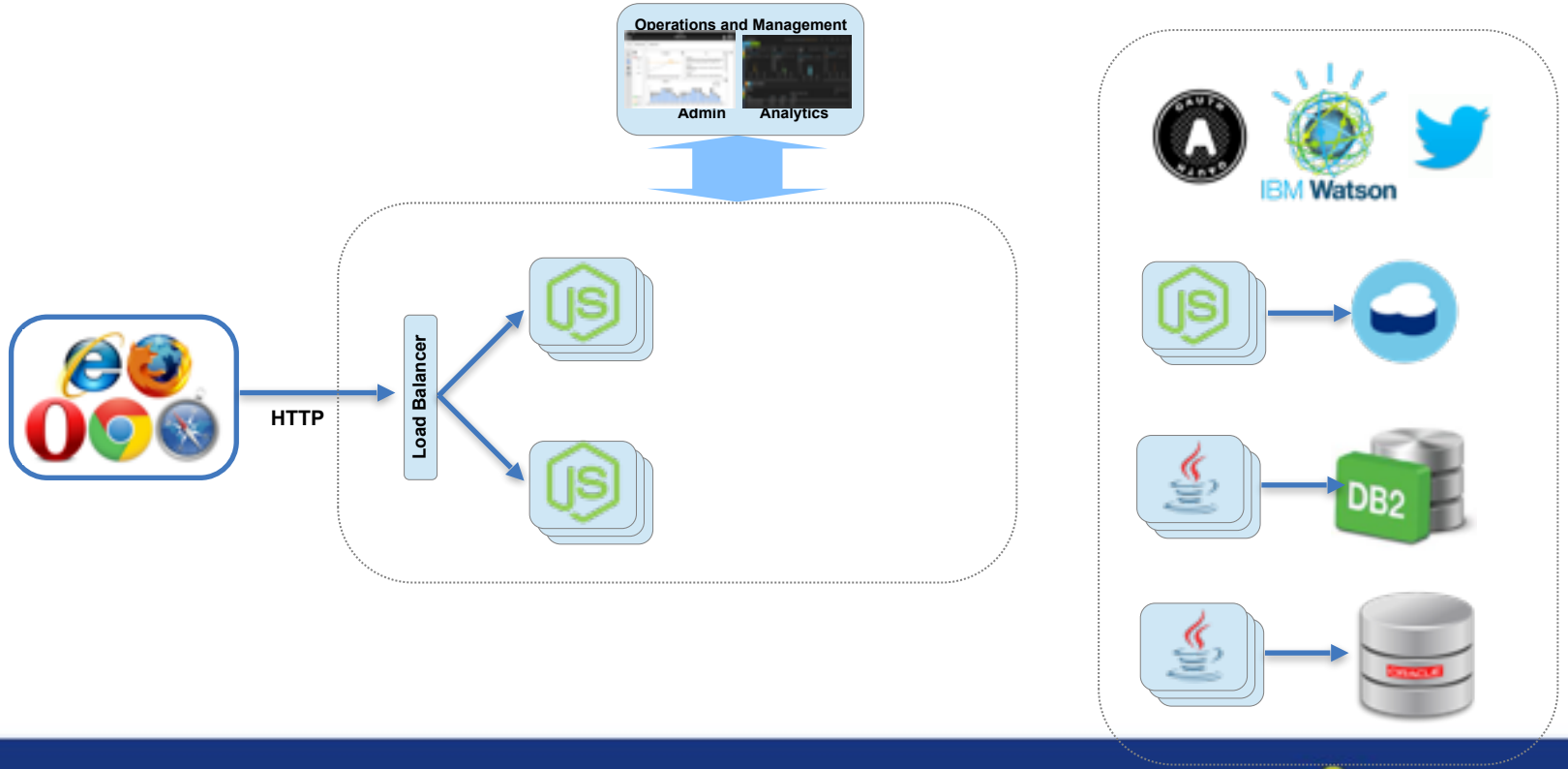
MicroServices and API Economy



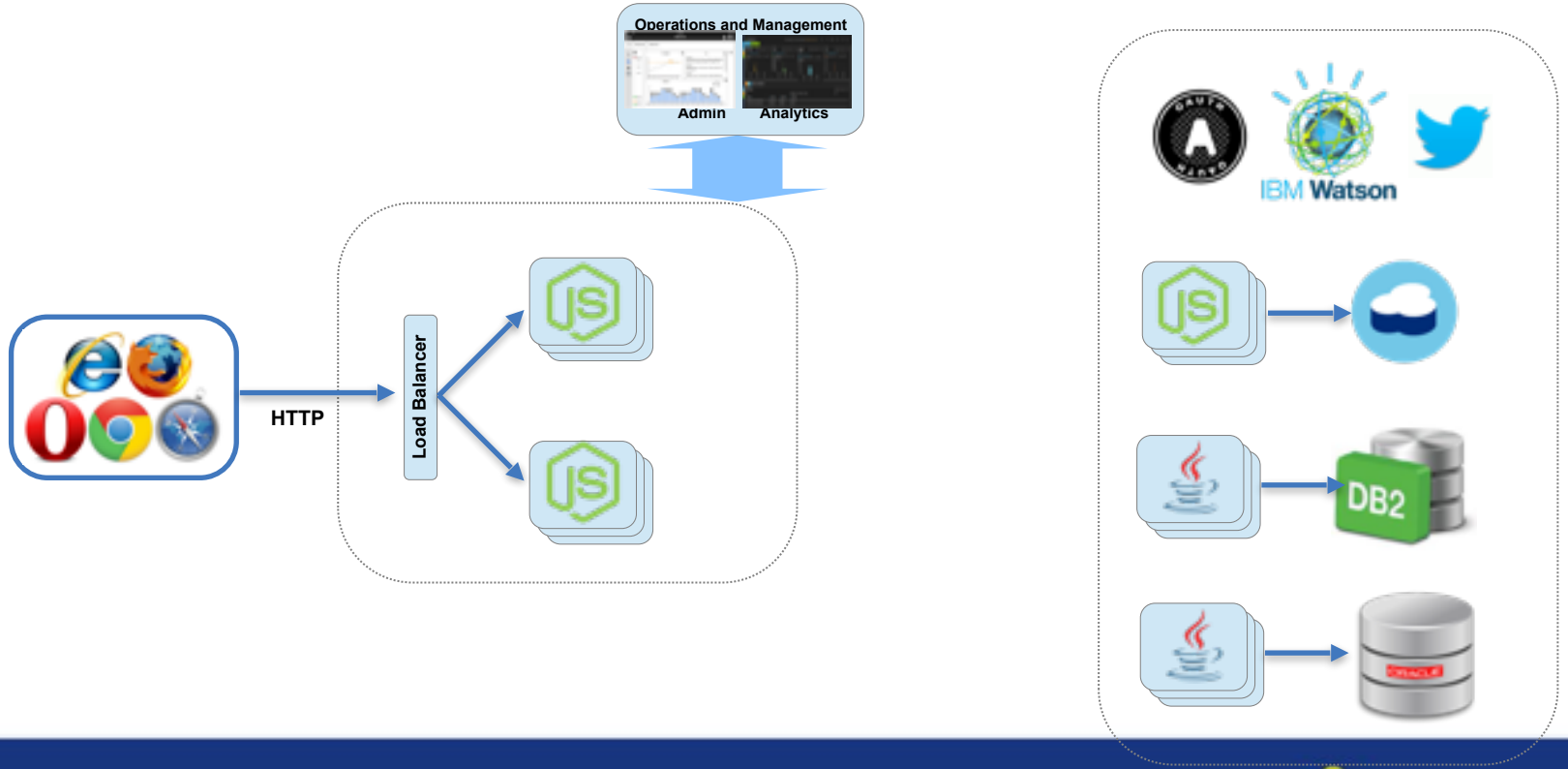
MicroServices and API Economy



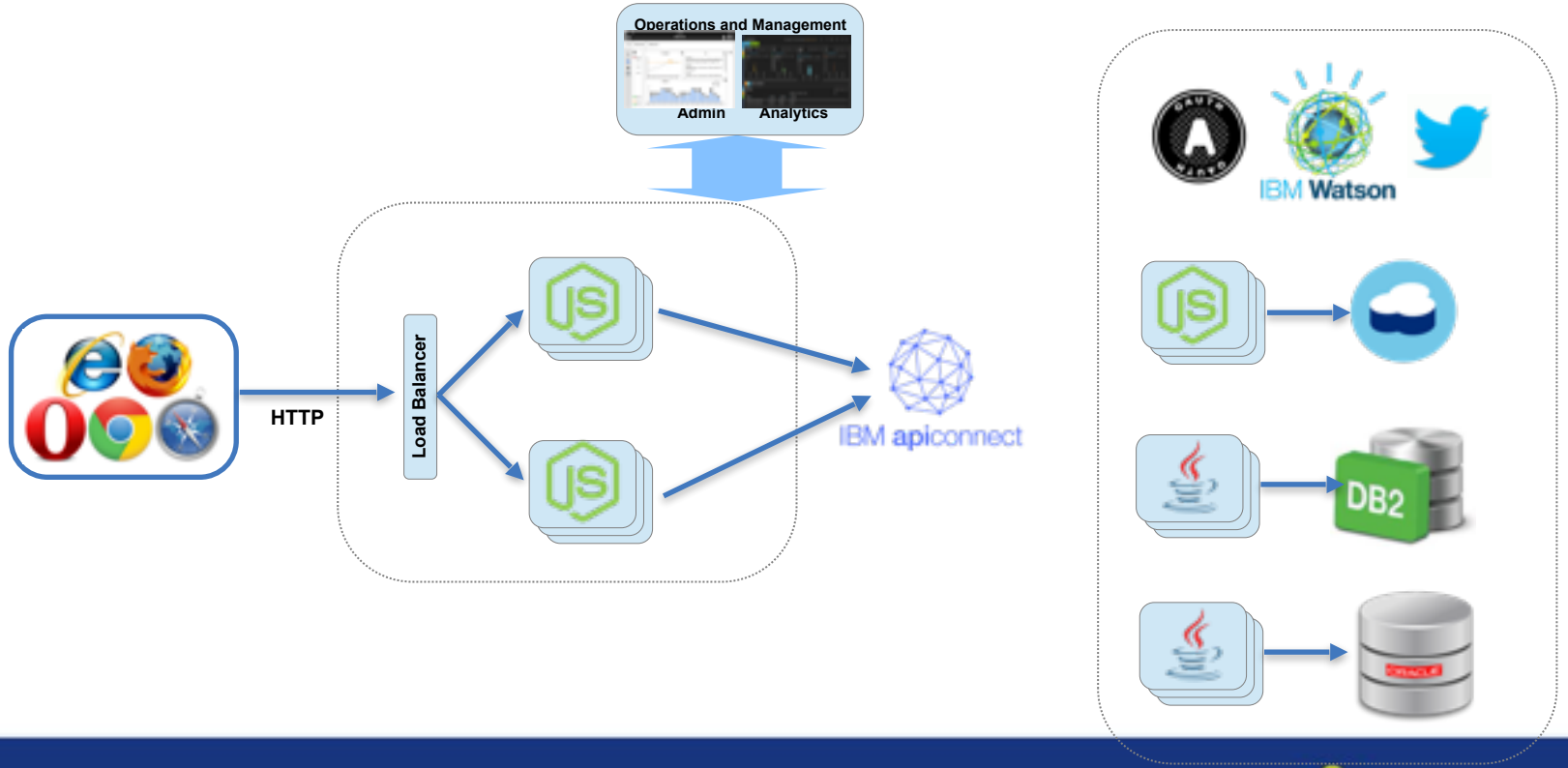
MicroServices and API Economy



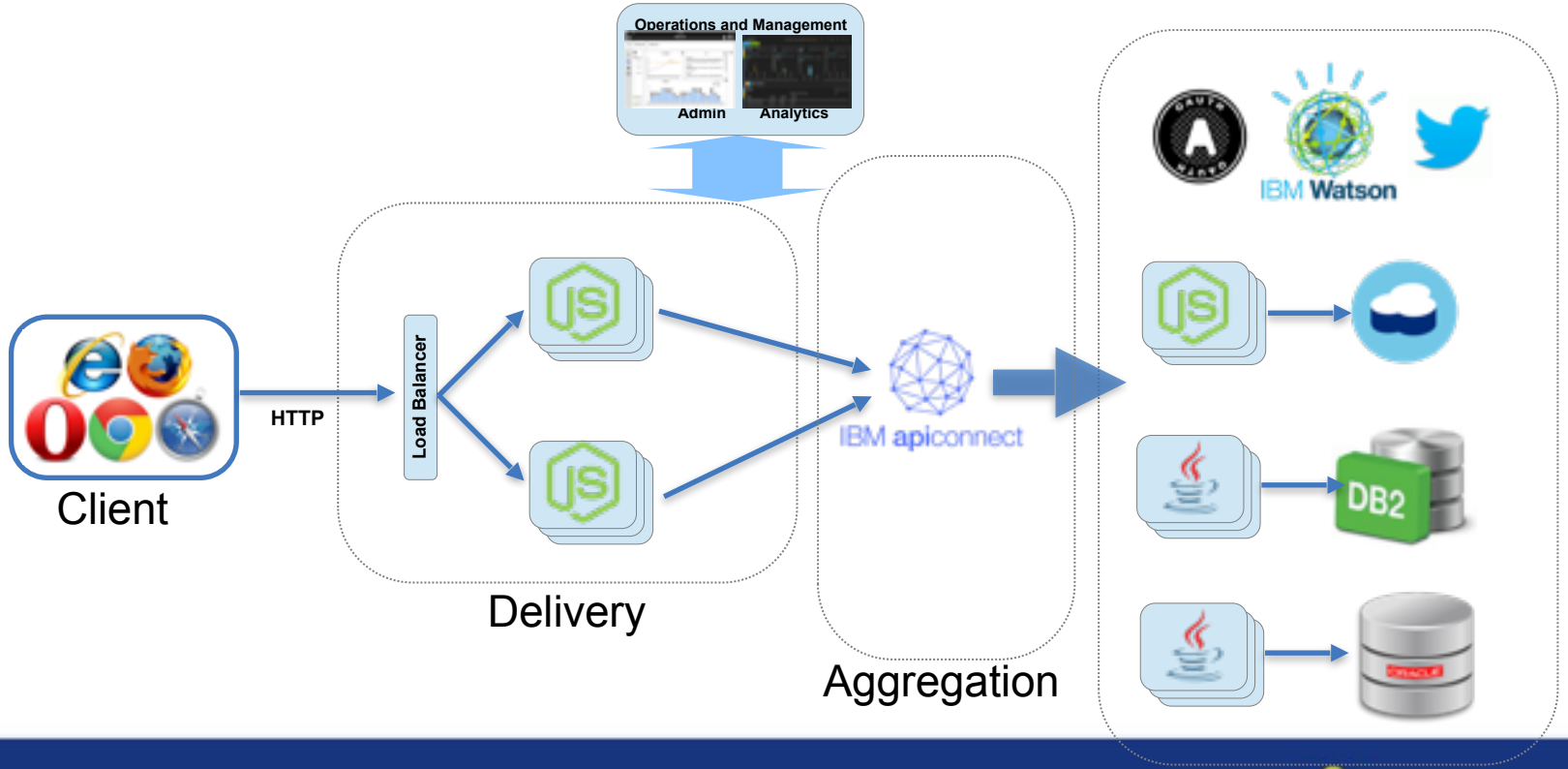
MicroServices and API Economy



MicroServices and API Economy



MicroServices and API Economy



Questions?

Thank You