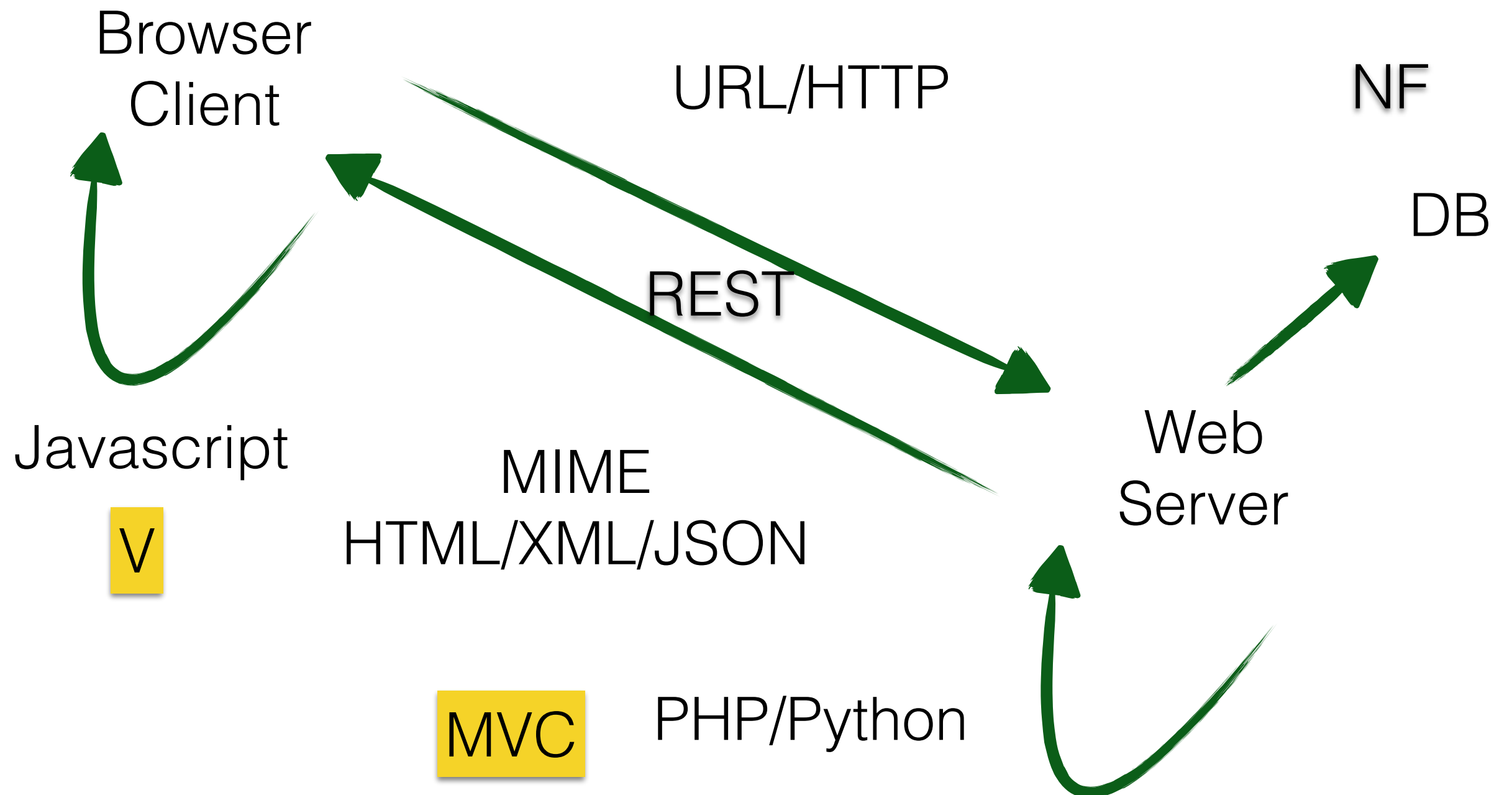


Web dynamic MVC + C/S Rendering

Frédéric Le Mouël & Julien Ponge

The Big Picture



People

- name
- email
- age



Edition Forms

Individual Sheet

PDF

Contact List

VIEW

Edition Forms

Individual Sheet

PDF

Contact List

ConTR0LLer



People

- name
- email
- age

MODEL

Separates Concerns

Business Code

MODEL

UPDATES

MANIPULATES

User Interactions

VIEW

CONTROLLER

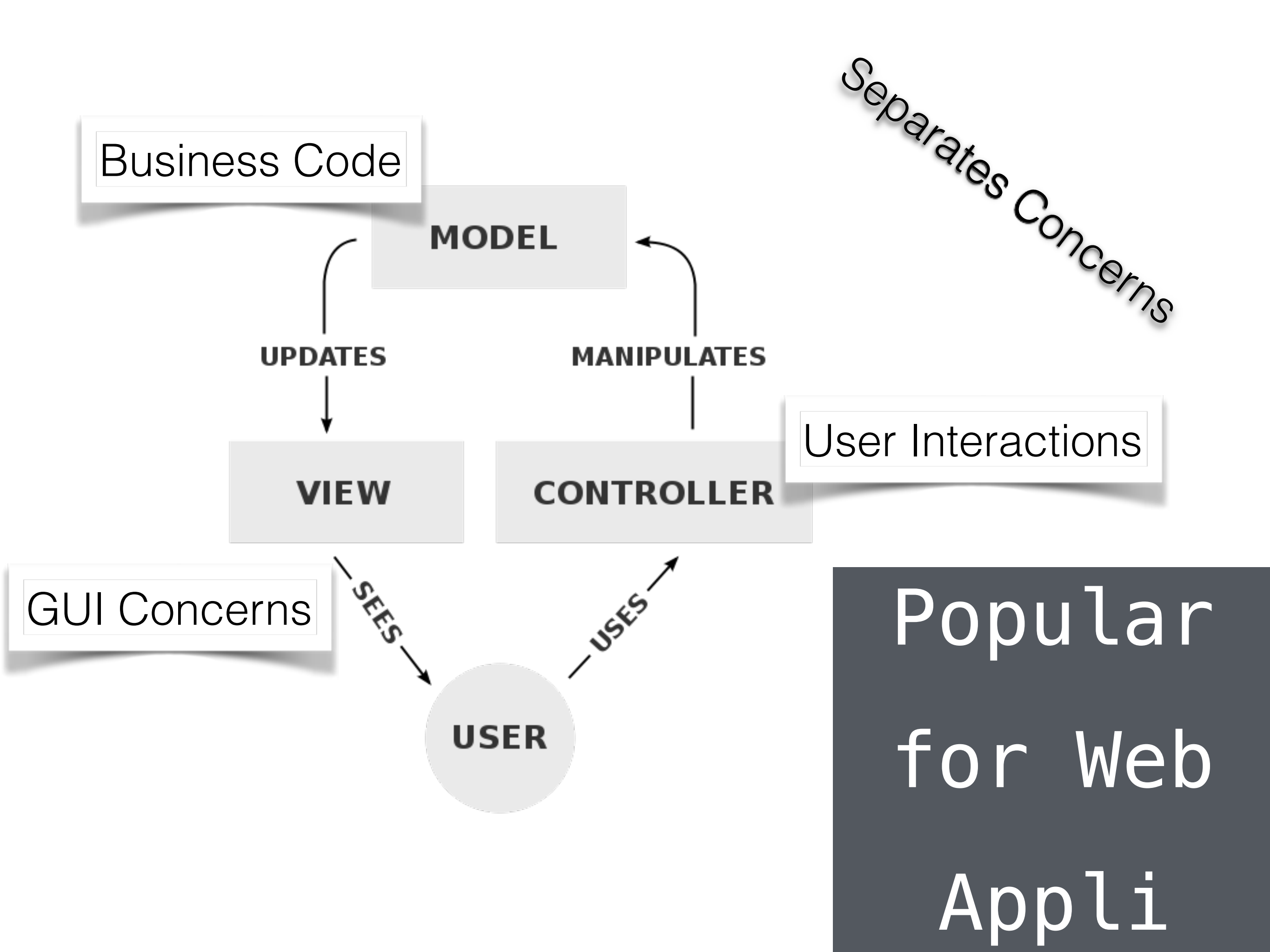
GUI Concerns

SEES

USES

USER

Popular
for Web
Appli

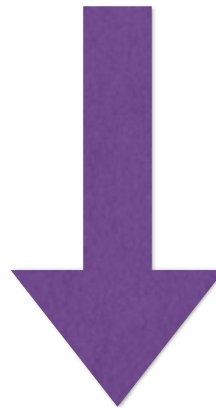


```
<!DOCTYPE HTML>
<html>
<head><title>Macaronic Code</title></head>
<body>
  <?php
    $ip = $_SERVER['REMOTE_ADDR'];
    $dbh = new PDO('mysql:host=localhost;dbname=mydb',
'username', 'password');
    $statement = $dbh->query("SELECT lastlog FROM logins WHERE
ip = '$ip'");
    $row = $statement->fetch(PDO::FETCH_ASSOC);
    $dbh->query("INSERT INTO logins VALUES ('$ip',
CURRENT_TIMESTAMP) ON DUPLICATE KEY UPDATE lastlog =
CURRENT_TIMESTAMP");
  ?>
  <p>Hello, <?php print $ip; ?>.</p>
  <script>
    var lastlog = "<?php print $row['lastlog']?>";
    if (lastlog > "") alert("You last visited "+lastlog);
    else alert("You've never been here before.");
  </script>
</body>
</html>
```

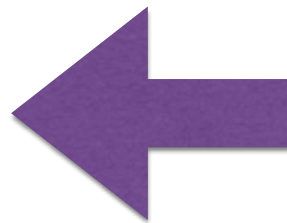
GET /contact/edit/15



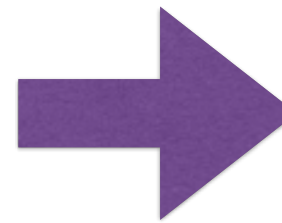
HTTP Routing



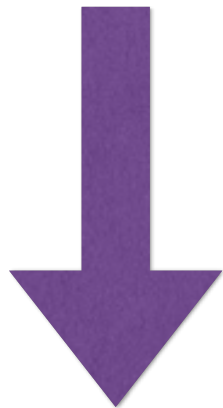
Data Model



Controller



HTML Rendering
(or else ...)



Persistence
(DB, files, ...)



GET

http://foo.com/**contacts**/**edit**/**15**

URL Mapping



“Controller”

```
1 @app.route('/contacts/<action>/<id>', methods=['GET', 'HEAD'])
2 def contact(action, id):
3     if action is 'edit':
4         contact = contact_from_db(id)
5         return render_template('edit.html', data=contact)
6     else:
7         # (...)
```

“View”
call

“Model”
call

ConTR0LLer

Database connection

```
1 def contact_from_db(id):  
2     db = engine.connect()  
3     row = db.execute(select([pages.c.text]).where(pages.c.id ==  
id)).fetchone()  
4     db.close()  
5     if row is None:  
6         return '**(This page is empty or does not exist.)**'  
7     return row[0]
```

Algorithmic
Data processing

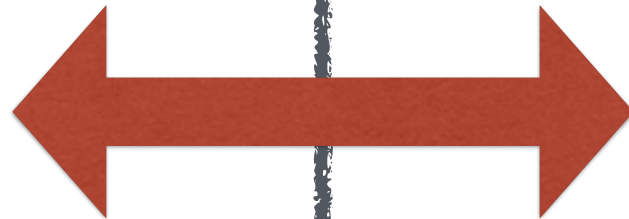
Data returned to the Controller

MODEL

Rendering server-side



HTML
CSS
JS
PNG
(...)

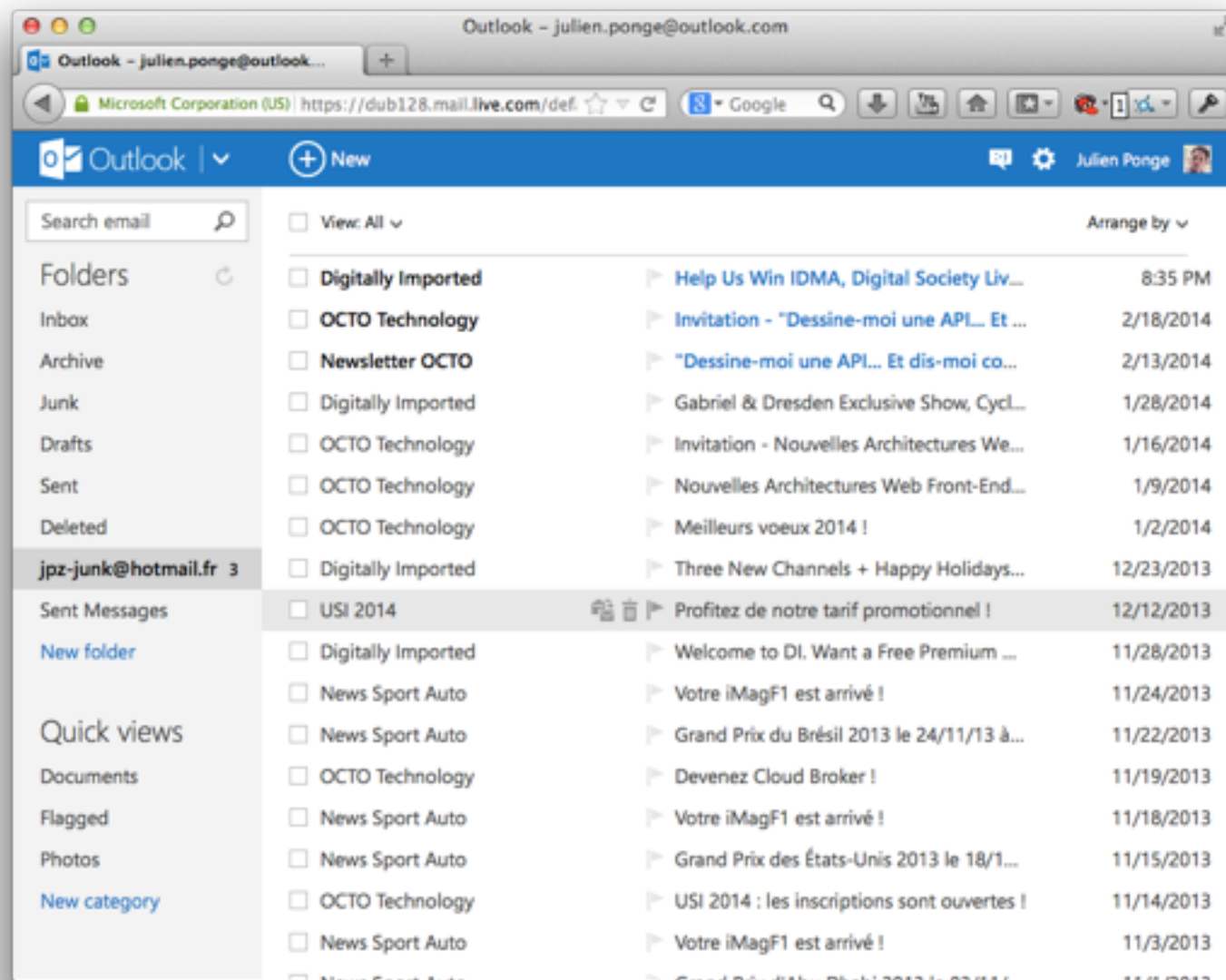


Classic
Static - if high traffic -
Caching
Page Reloading

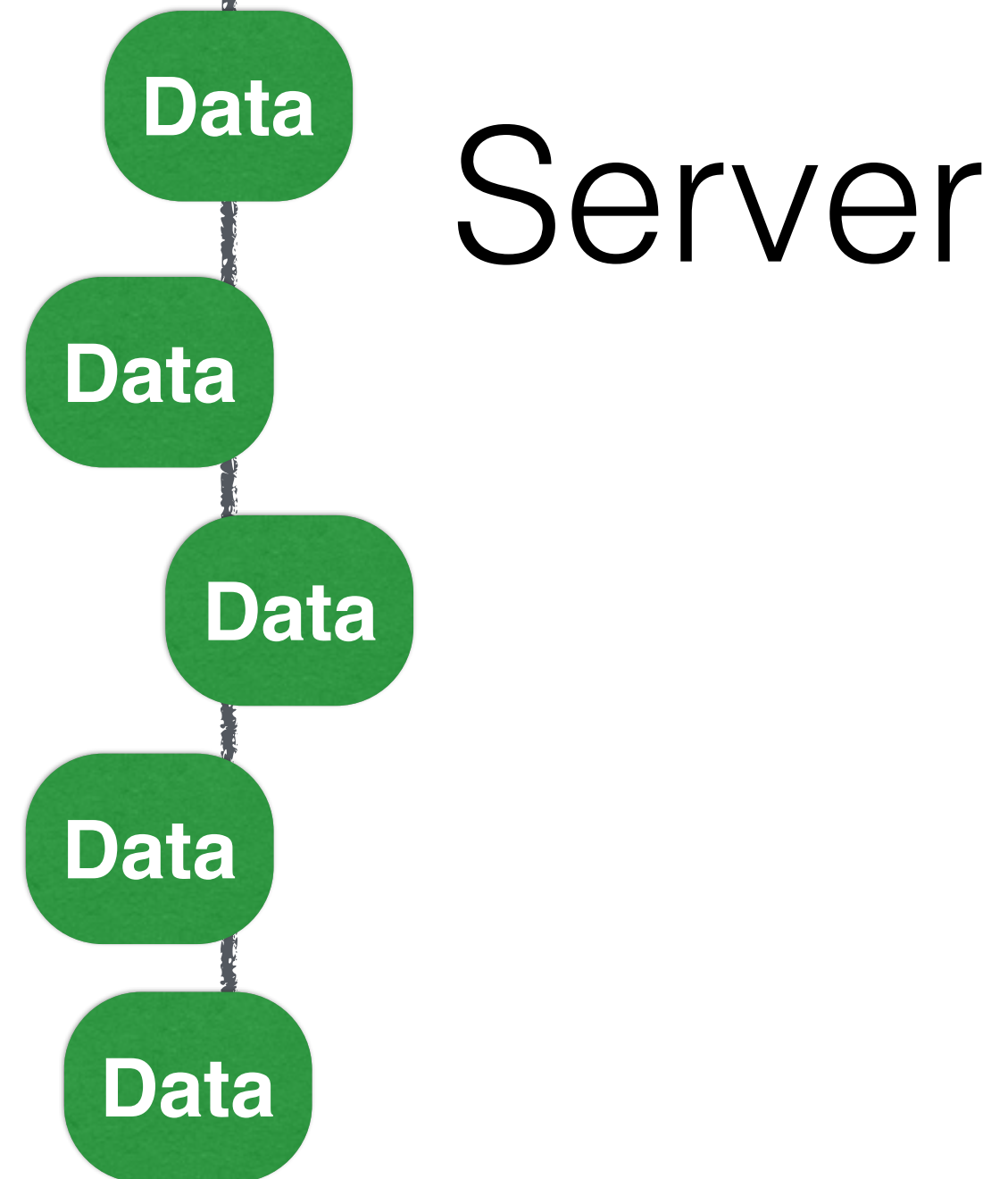
Templating

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Contacts</title>
</head>
<body>
  <h1>Contacts</h1>
  <ul>
    {% for contact in data %}
      <li>{{ contact['name'] }} ({{ contact.email }})</li>
    {% endfor %}
  </ul>
  <form action="/update">
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

User interaction



Rendering
Client-side



1

Statique

Bootstrap

HTML

CSS

JS

2 3 ...

Pull / Push

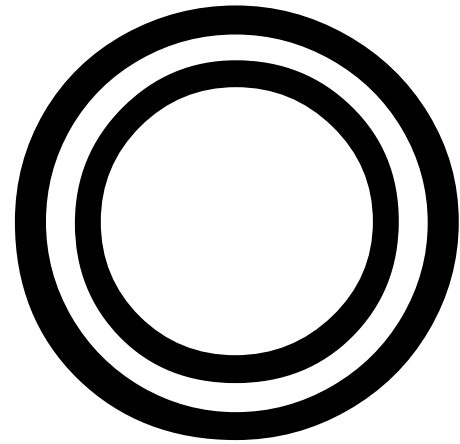
async Request

HTTP

JSON

HTML

CSS



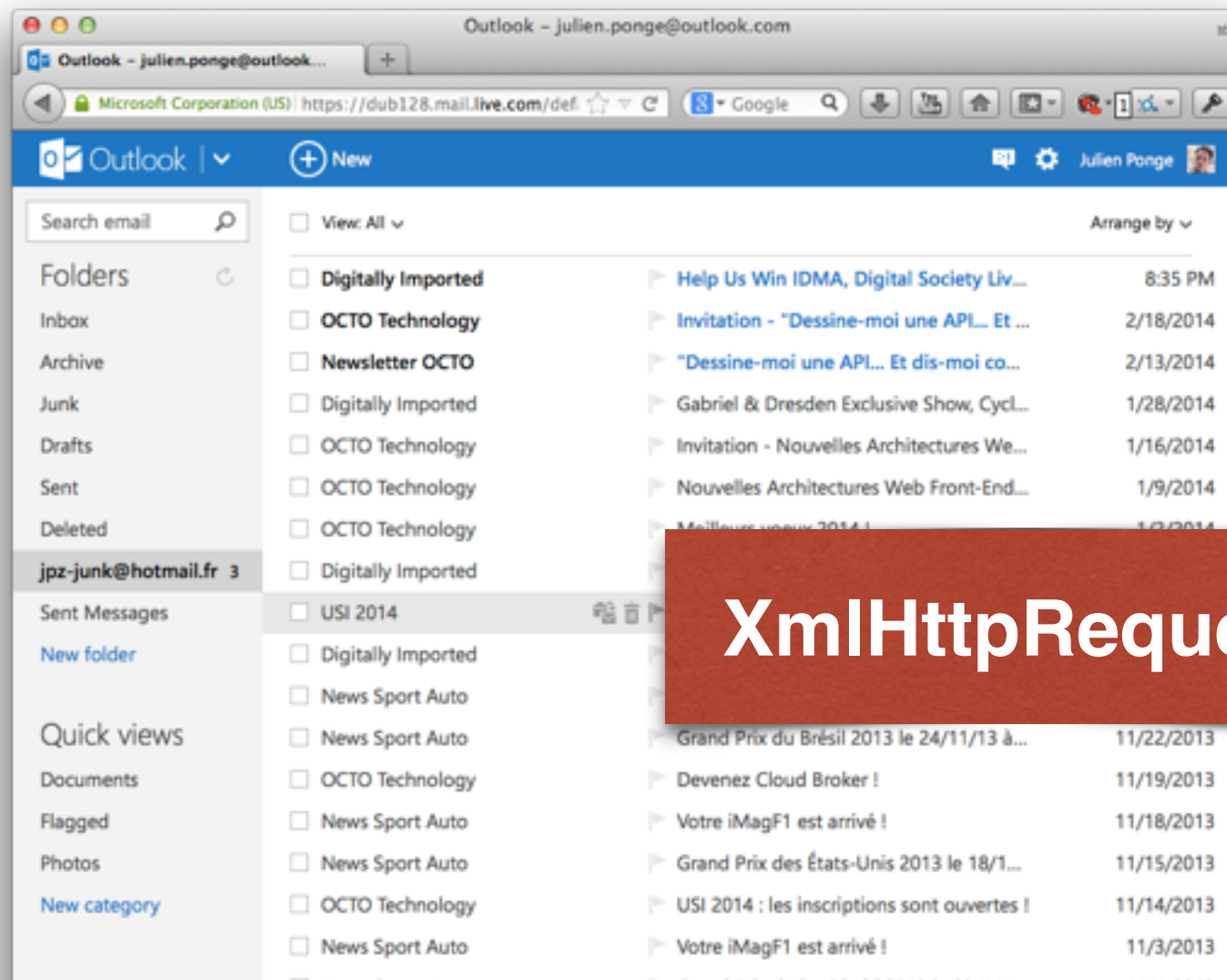
JavaScript

```
{  
  'contact': {  
    'name': 'Mr Bean',  
    'email': 'mrbean@bean.com',  
    'friends': [  
      'Mary', 'John'  
    ]  
  }  
}
```

JavaScript Object Notation (JSON)

```
{  
  'contact': {  
    'name': 'Mr Bean',  
    'email': 'mrbean@bean.com',  
    'friends': [  
      'Mary', 'John'  
    ]  
  }  
}
```

JSON.stringify(obj) -> string
JSON.parse(string) -> obj



XmlHttpRequest

JSON

Server

JSON

JSON

```
$.ajax({  
  url: "/api/sum/" + a + "/" + b,  
  success: function(data) {  
    alert(data.result);  
  },  
  error: function() {  
    alert("Fail!");  
  }  
});
```

```
$.ajax({  
  type: 'POST',  
  url: '/api/register',  
  data: JSON.stringify({  
    'login': form.login,  
    'password': form.password,  
    'country': 'FRA'  
  }),  
  dataType: 'json',  
  contentType: 'application/json; charset=utf-8',  
  success: function(data) {  
    // (...)  
  },  
  error: function() {  
    // (...)  
  }  
});
```

/contacts

/contact/10

Create



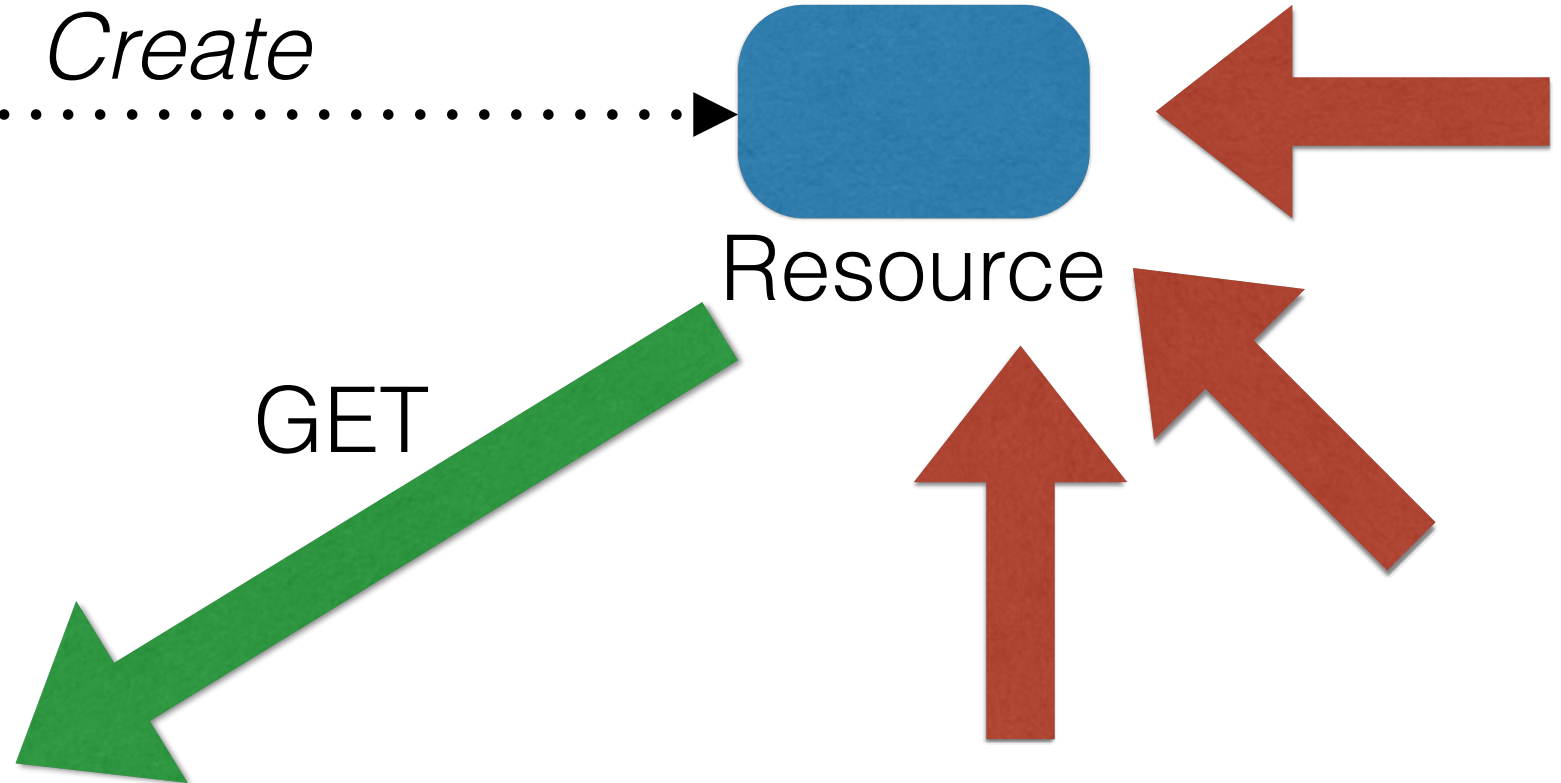
POST

GET

Resource

PUT, DELETE, PATCH, ...

```
{  
  'contact': {  
    'name': 'Mr Bean',  
    'email': 'mrbean@bean.com',  
    'friends': [  
      'Mary', 'John'  
    ]  
  }  
}
```





Perfect for business applications
No page reloading
Unload the server



Complex client code
Bad research engine indexing