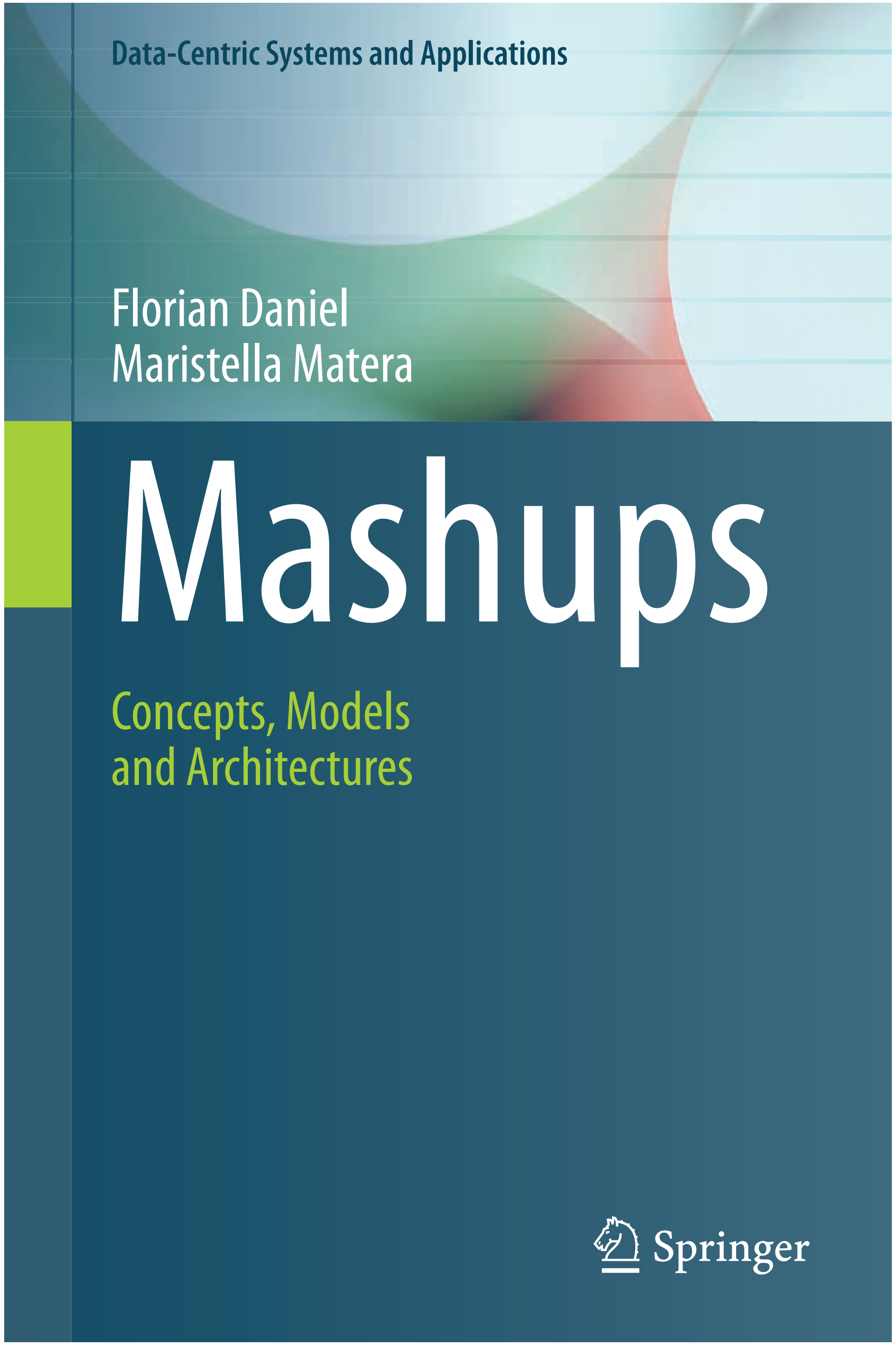

Chapter 3
Web Technologies

Figures

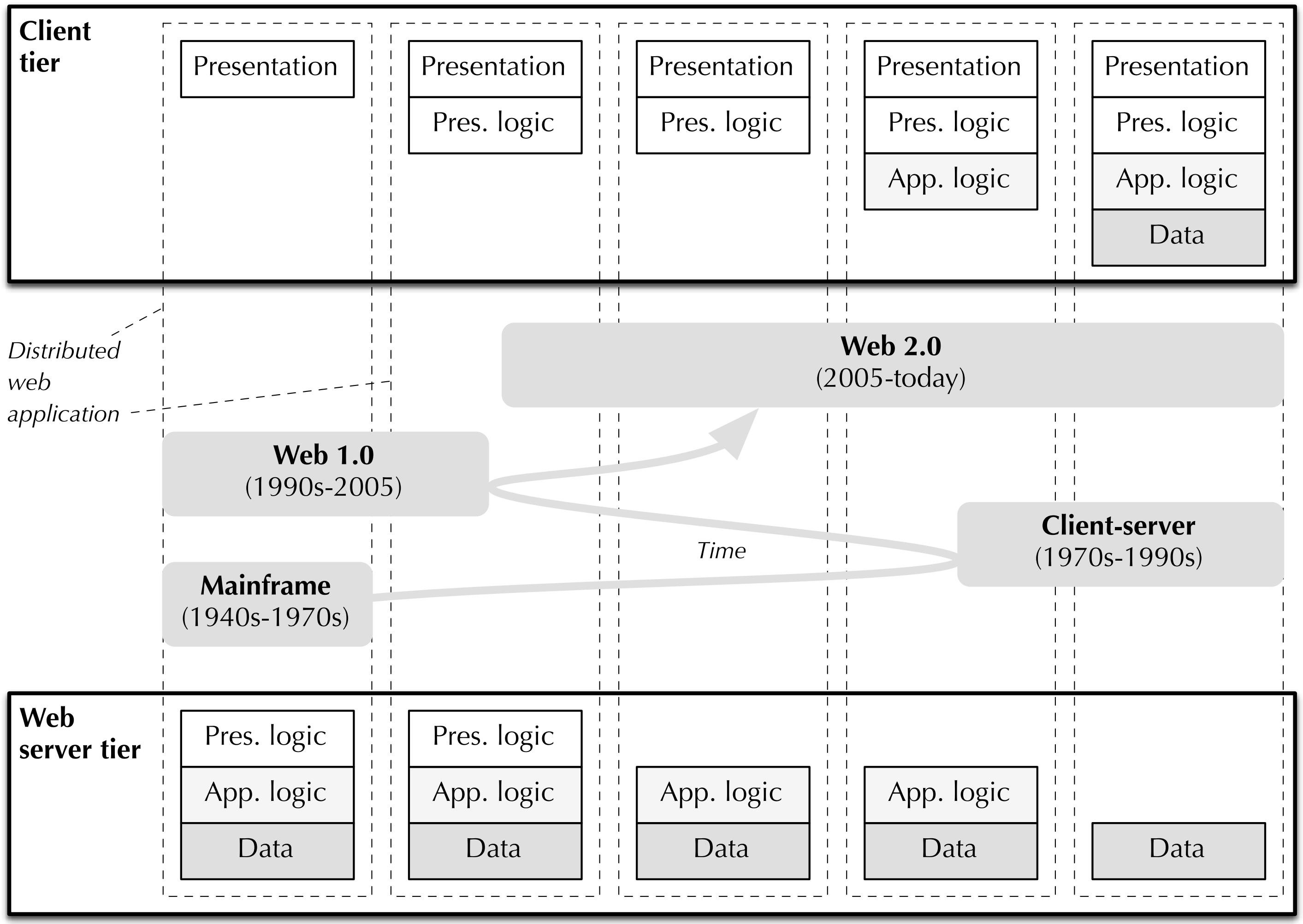

Fig. 3.1 The change of the distribution of a web application's internal architectural layers over client and server over time (adapted from http://www.coachwei.com).

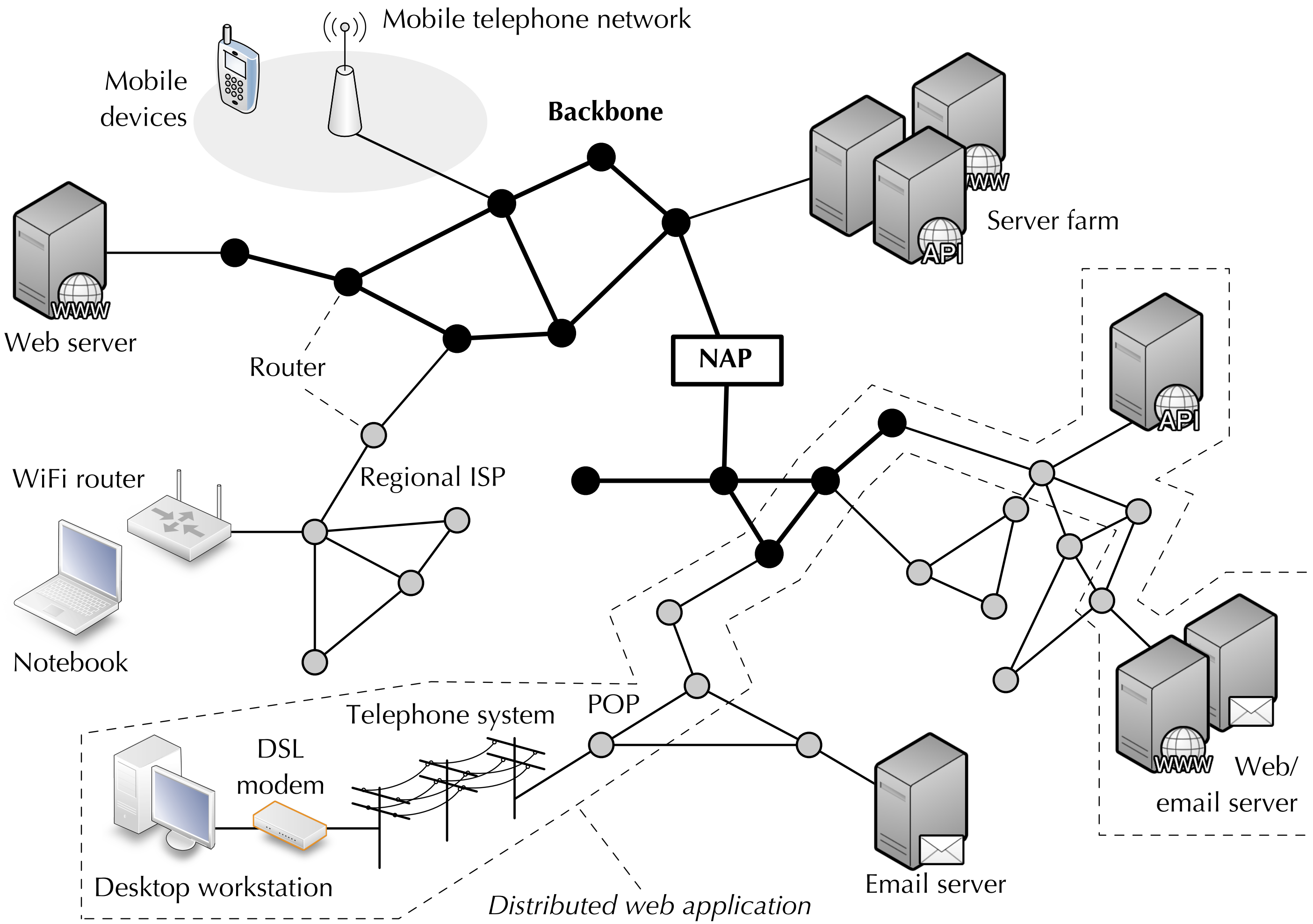

Fig. 3.2 Internet architecture (adapted from [259]). The dashed polygon describes a possible distribution of a web application over the Internet.

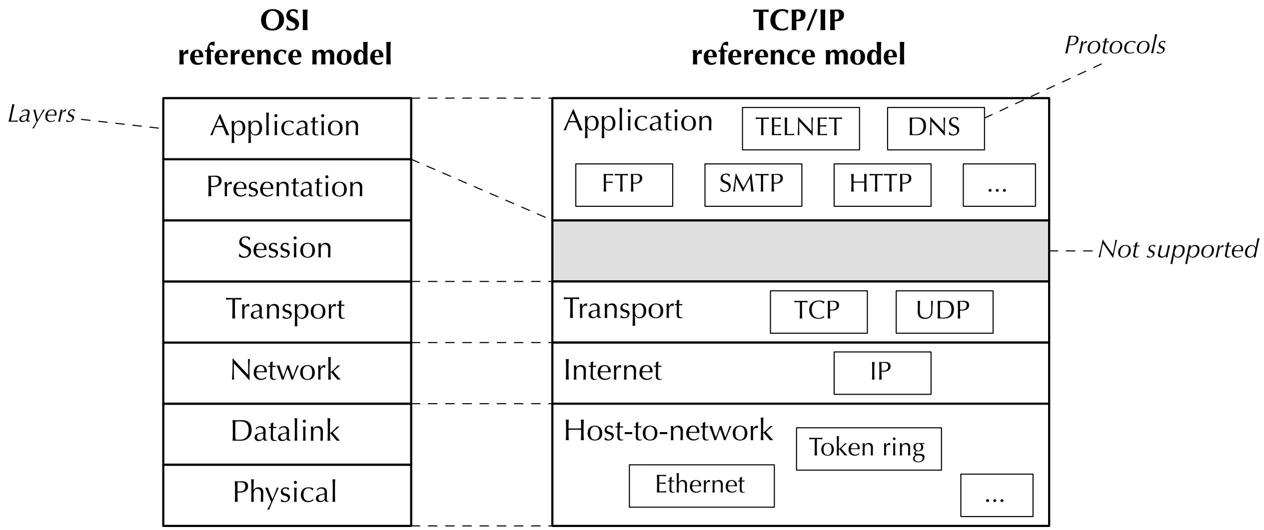

Fig. 3.3 The TCP/IP reference model compared to the OSI reference model.

```
<!DOCTYPE html>
<HTML>
<HEAD>
      <TITLE>Expo 2015 - An example</TITLE>
      <meta name="description" content="An HTML example">
      <meta name="Keywords" content="HTML">
</HEAD>

<BODY>

<div id="top">
      <img src="Logo_EXPO.png" ALT="EXPO official logo"/>
</div>

<div id="center">
      <h1>A simple Web page linking to the Expo2015 Web site</h2>
      <p>We here show an example of image and video inclusion and of <a
href="http://www.worldexpo2015.it/index.html"> link definition</a></p>
</div>

<div id="youtube" class="container vertical">
    <iframe id="video_canvas" src="http://www.youtube.com/embed/mSbyzSJl-
      eM?list=PL3IPVSF68uaEfHAVNek3RBb1w4P05KiHi" width="800"
      height="500"frameborder="0" allowfullscreen></iframe>
</div>
<div id="bottom">

</BODY>
</HTML>
```

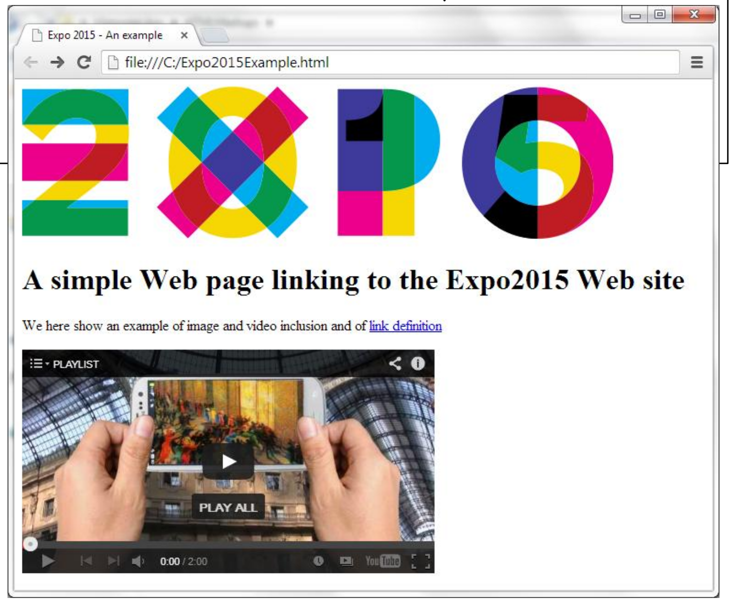

Fig. 3.4 A simple HTML page embedding an image and a video and including a clickable hyperlink.

```
<html>
<head>
<title>EXPO 2015</title>
<script type="text/javascript">
        var nextButton;
        var videoCanvas;
        var videoArray;
        var currentVideo = -1;

        function nextVideo() {
                currentVideo++;
                if(currentVideo >= videoArray.length)
                        currentVideo = 0;
                videoCanvas.setAttribute("src", videoArray[currentVideo]);
        }

        function init() {
                nextButton = document.getElementById("next_button");
                videoCanvas = document.getElementById("video_canvas");
                videoArray = [
                        "http://www.youtube.com/embed/kNG_l_UKkgM",
                        "http://www.youtube.com/embed/JdK1bIg1VvA",
                        "http://www.youtube.com/embed/m_F8A5VdhsM"
                ];
                nextButton.addEventListener("click", nextVideo, false);
                nextVideo();
        }
</script>
</head>
<body onload="init()">
/* mark-up for other page elements */
        <p><a id="next_button" href="javascript:nextVideo()">Next video</a></p>
        <iframe id="video_canvas" width="800" height="500" src="" frameborder="0"
allowfullscreen></iframe>
</body>
</html>
```

Fig. 3.5 An example of JavaScript code included in the head section of an HTML page. The script adds interactivity to the page, by implementing a video slide show.

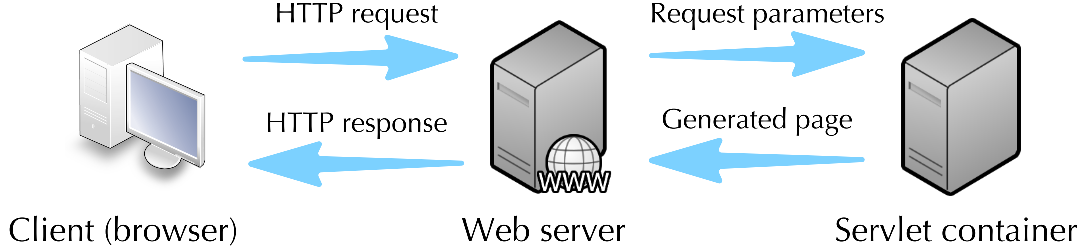

Fig. 3.6 Java servlet architecture.

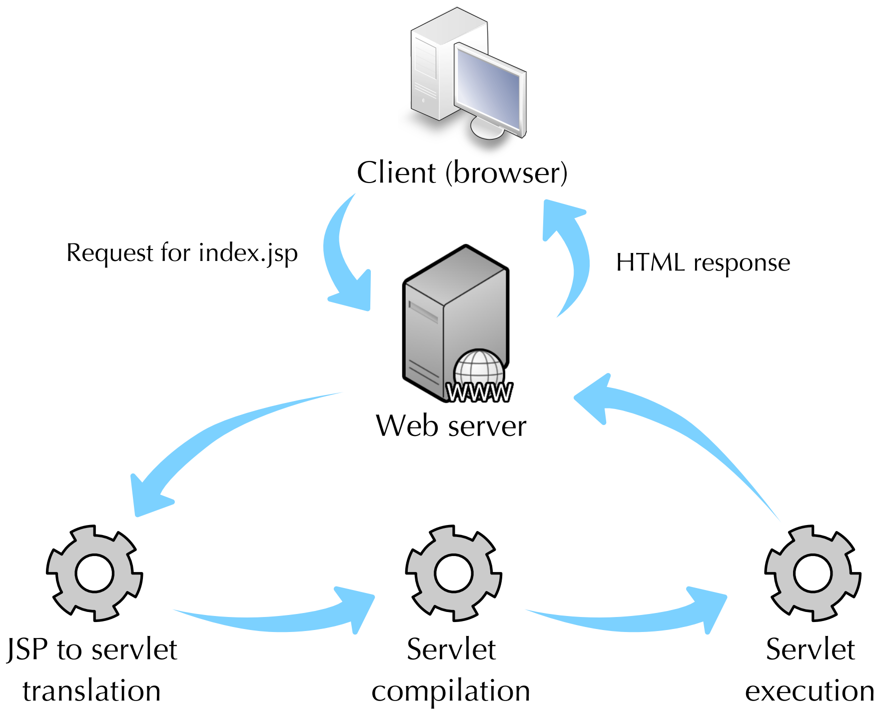

Fig. 3.7 The translation of JSP pages into servlets.

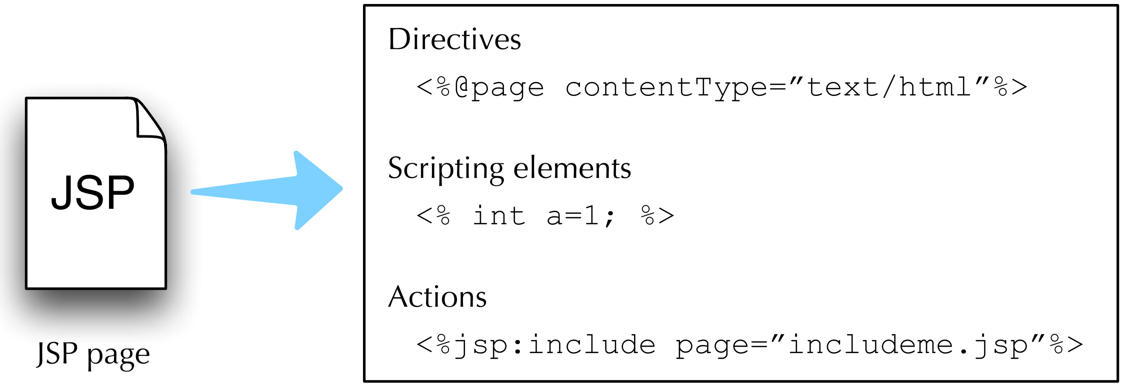

Fig. 3.8 The structure of JSP pages.

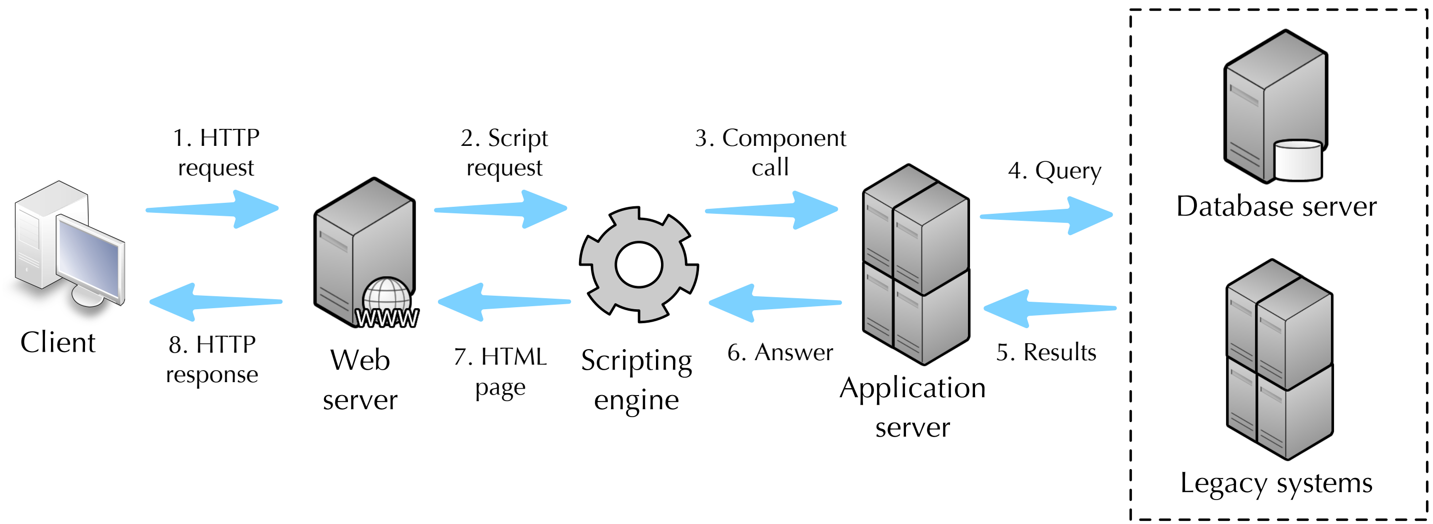

Fig. 3.9 Application server architecture

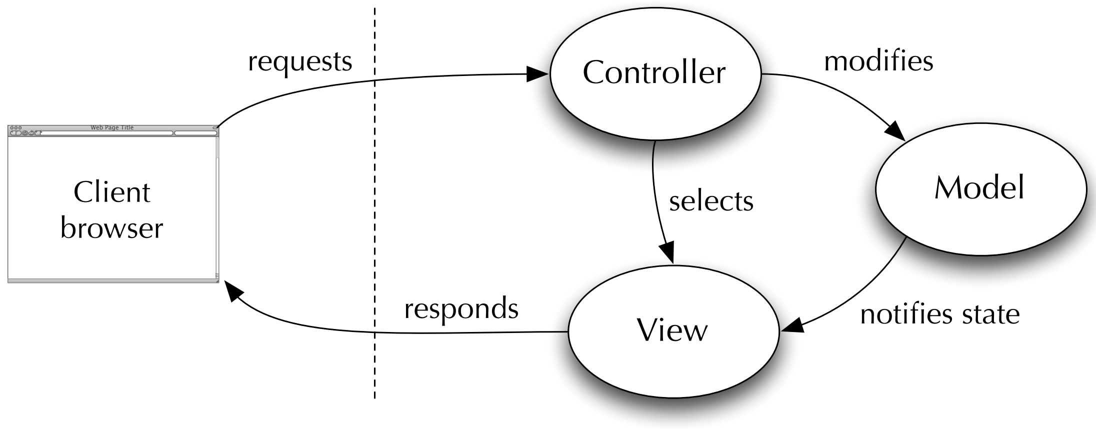

Fig. 3.10 The Model-View-Controller pattern adapted to Web applications.

JSON

```
{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}}
```

XML

```
<menu id="file" value="File">
  <popup>
    <menuitem value="New" onclick="CreateNewDoc()" />
    <menuitem value="Open" onclick="OpenDoc()" />
    <menuitem value="Close" onclick="CloseDoc()" />
  </popup>
</menu>
```

Fig. 3.11 Examples of JSON and XML specifications. The represented data refer the structure of a file menu (http://www.json.org/example)