

Managing State

Chapter 16

Chapter 16

1

The Problem of
State in Web
Applications

2

Passing
Information via
Query Strings

3

Passing
Information via
the URL Path

4

Cookies

5

Serialization

6

Session State

7

HTML5 Web
Storage

8

Caching

Chapter 16 cont.

9

Summary

Chapter 16

1

The Problem of
State in Web
Applications

2

Passing
Information via
Query Strings

3

Passing
Information via
the URL Path

4

Cookies

5

Serialization

6

Session State

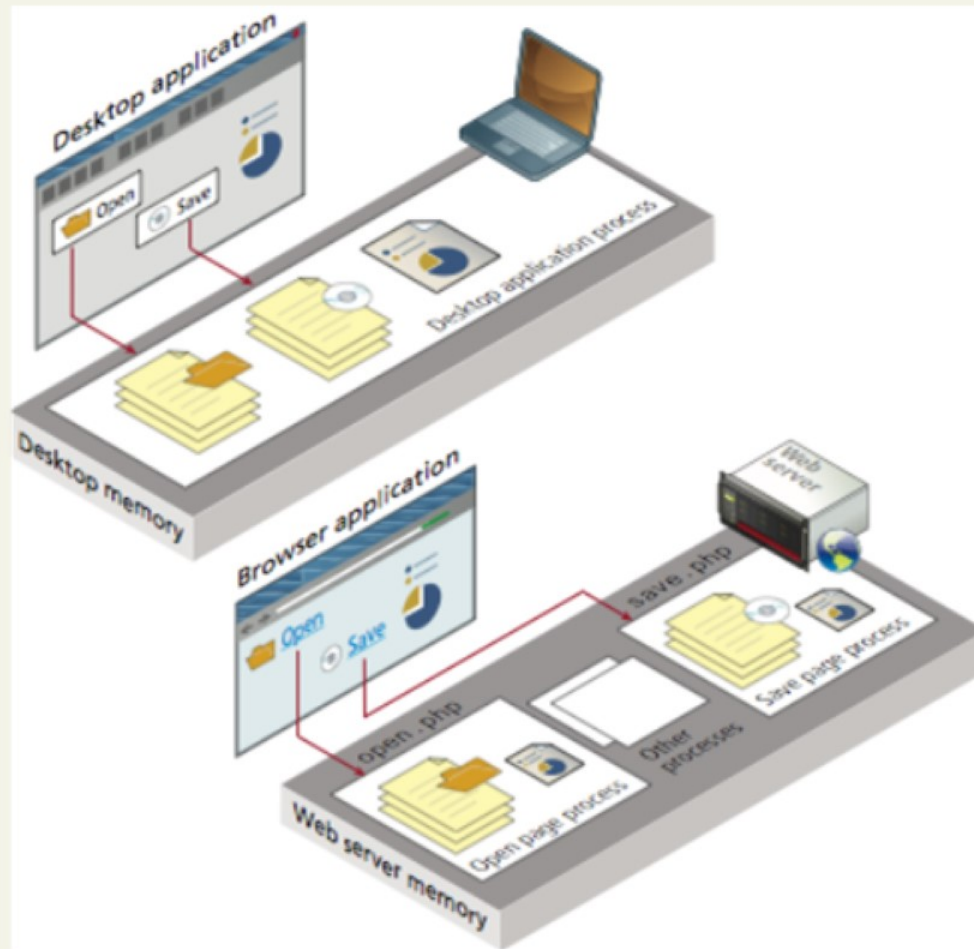
7

HTML5 Web
Storage

8

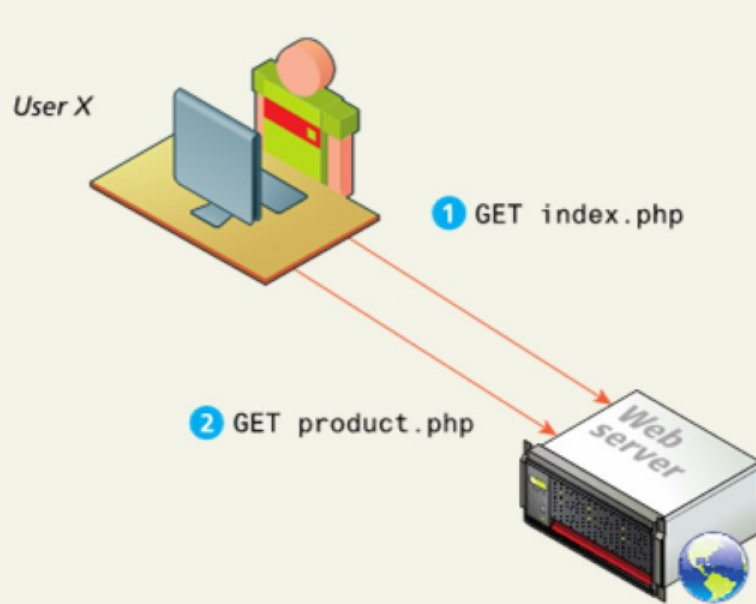
Caching

The Problem of State in Web Applications

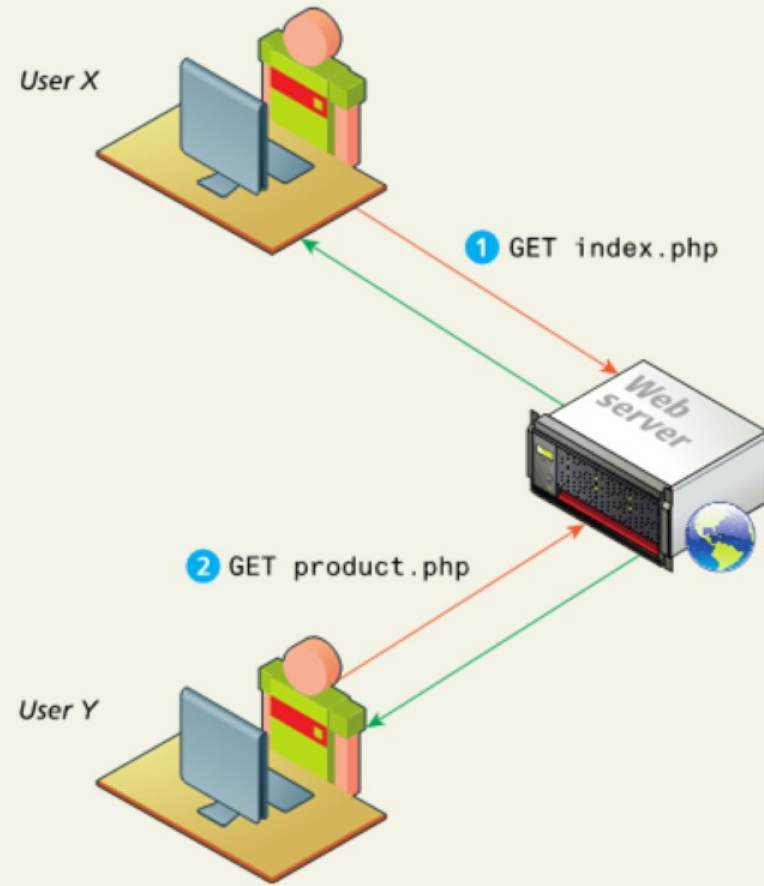


The Problem of State in Web Applications

What the server sees

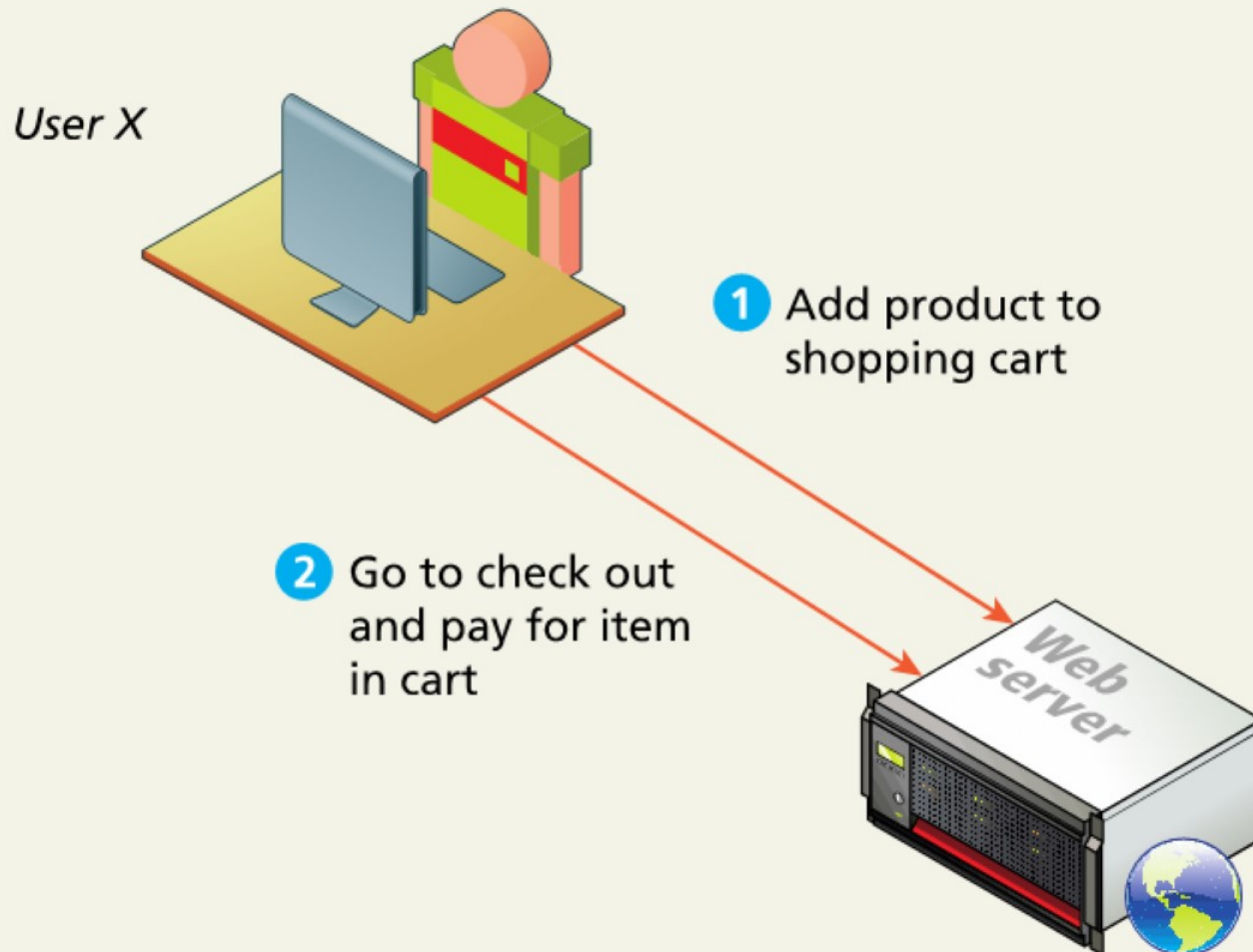


... is for the server not really any different than ...



The Problem of State in Web Applications

What the user wants



The Problem of State in Web Applications

Solution

- Query Strings
- Cookies
- Sessions

Chapter 16

1

The Problem of
State in Web
Applications

2

Passing
Information via
Query Strings

3

Passing
Information via
the URL Path

4

Cookies

5

Serialization

6

Session State

7

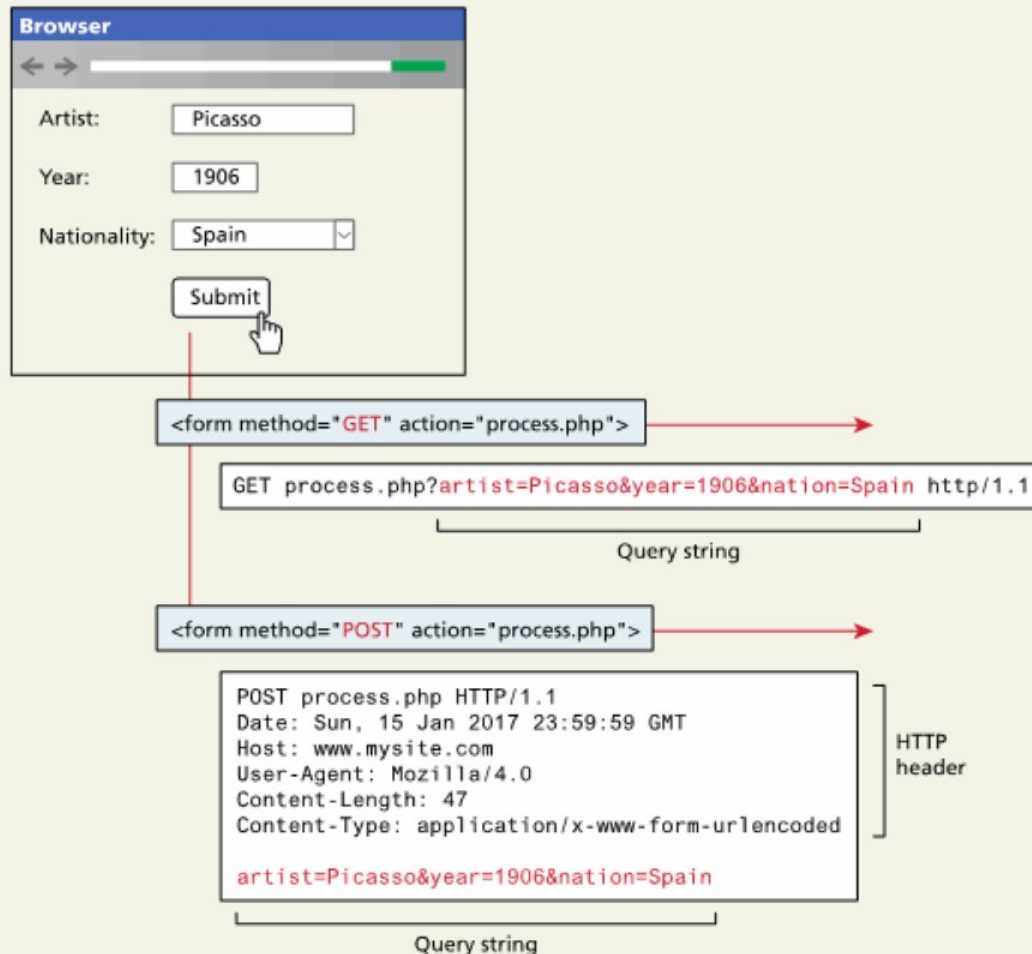
HTML5 Web
Storage

8

Caching

Passing Information via Query Strings

Remember POST and GET



Chapter 16

1

The Problem of
State in Web
Applications

2

Passing
Information via
Query Strings

3

Passing
Information via
the URL Path

4

Cookies

5

Serialization

6

Session State

7

HTML5 Web
Storage

8

Caching

Passing Information via the URL Path

Dynamic URLs (i.e., query strings) are a pretty essential part of web application development.

Static URLs

- the appearance of search terms within the URL does seem to improve its relative position.
- Another benefit to static URLs is that users tend to prefer them.
- Can use **URL rewriting** to make static URLs be dynamic

Passing Information via the URL Path

URL Rewriting in Apache and Linux

<http://www.1st-art-gallery.com/Raphael/La-Donna-Velata-1516.html> ←

<http://www.paintingall.com/raphael-sanzio-woman-with-a-veil-la-donna-velata.html>

<http://www.artsheaven.com/raphael-la-donna-velata.html>



<http://www.paintingswholesaler.com/detail.asp?vcode=6umd7krr1yqi161c&title=La+Donna+Velata>

Chapter 16

1

The Problem of
State in Web
Applications

2

Passing
Information via
Query Strings

3

Passing
Information via
the URL Path

4

Cookies

5

Serialization

6

Session State

7

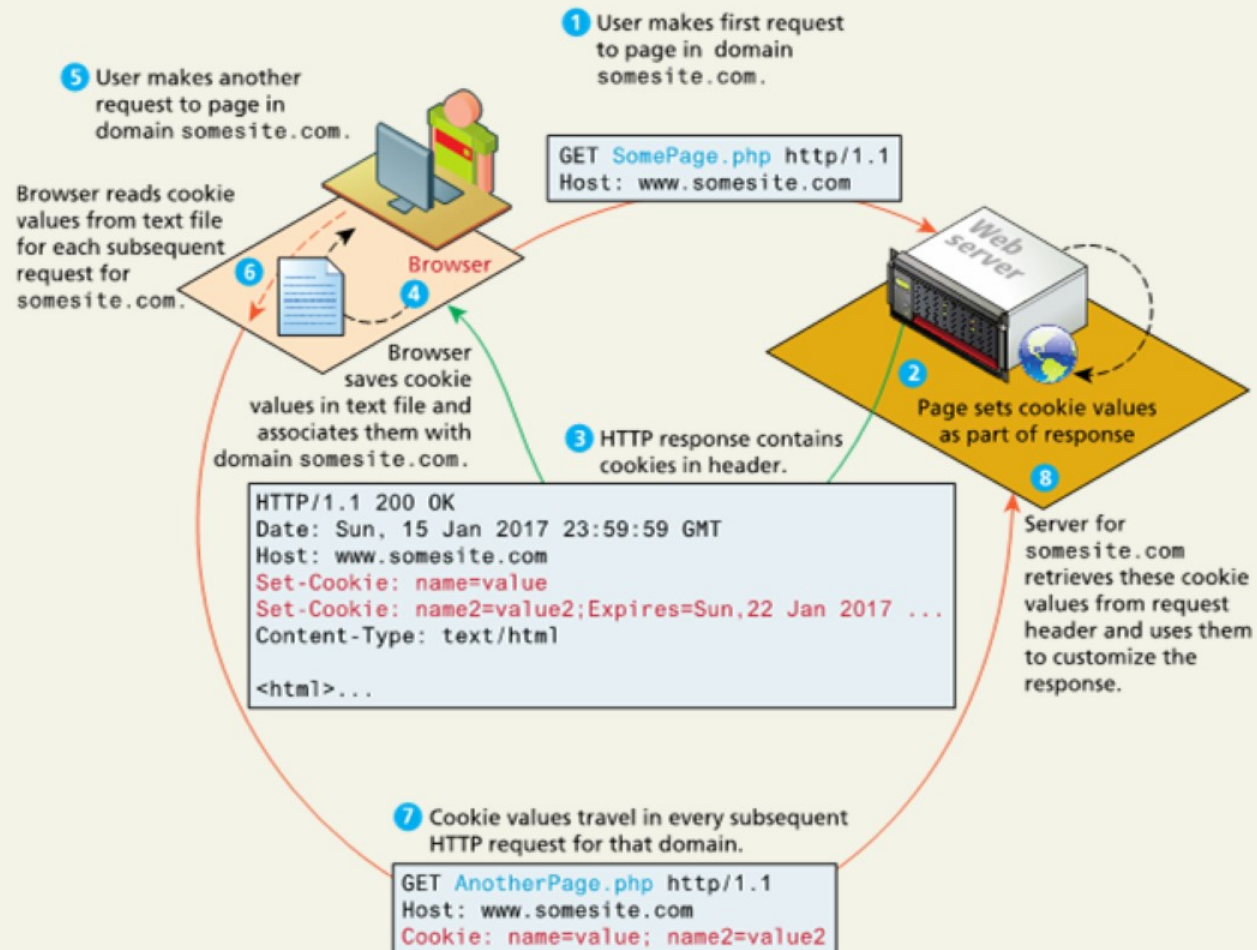
HTML5 Web
Storage

8

Caching

Cookies

How Do Cookies Work?



Cookies

Setting a Cookie

```
<?php  
  
// add 1 day to the current time for expiry time  
  
$expiryTime = time()+60*60*24;  
  
// create a persistent cookie  
  
$name = "Username";  
  
$value = "Ricardo";  
  
setcookie($name, $value, $expiryTime);  
  
?>
```


Cookies

Reading a Cookies

```
<?php
if( !isset($_COOKIE['Username']) ) {
    //no valid cookie found
}
else {
    echo "The username retrieved from the cookie is:";
    echo $_COOKIE['Username'];
}
?>
```

Cookies

Persistent Cookie Best Practices

Chapter 16

1

The Problem of
State in Web
Applications

2

Passing
Information via
Query Strings

3

Passing
Information via
the URL Path

4

Cookies

5

Serialization

6

Session State

7

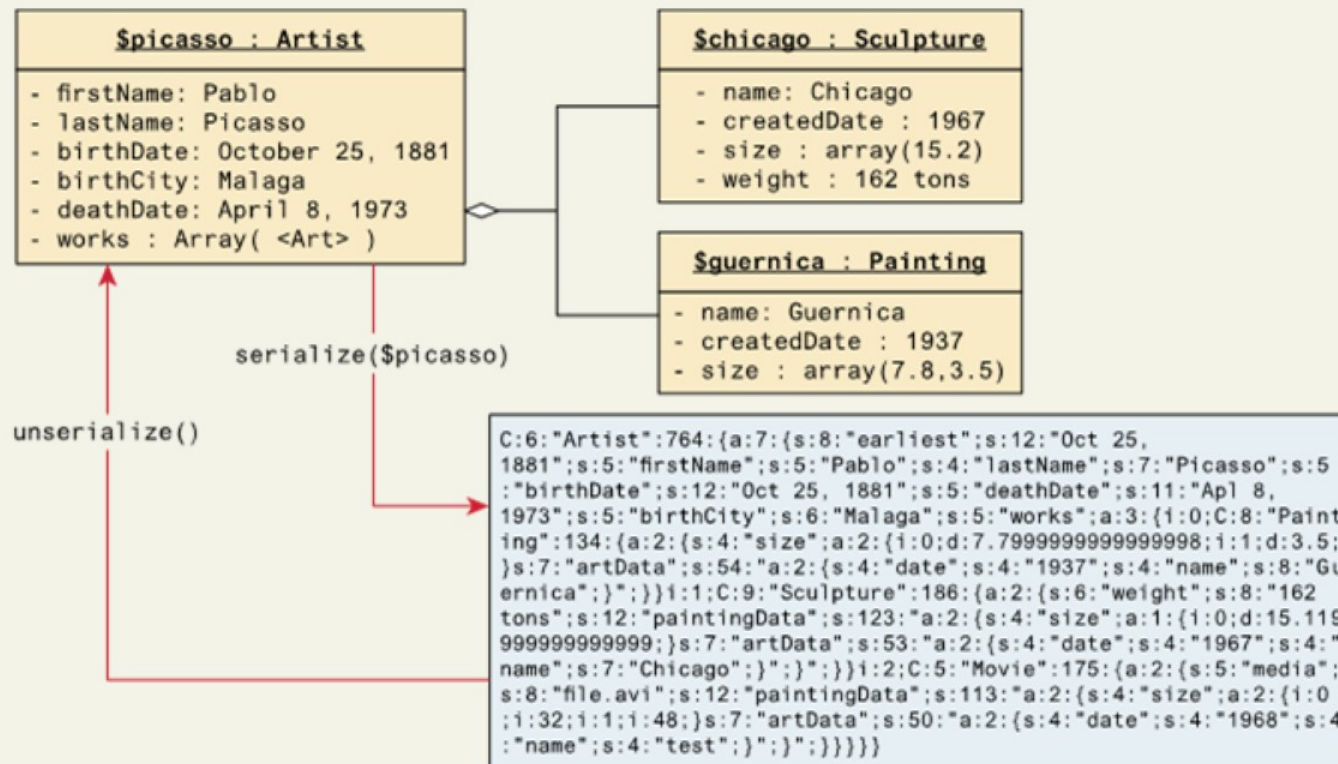
HTML5 Web
Storage

8

Caching

Serialization

Application of Serialization



Chapter 16

1

The Problem of
State in Web
Applications

2

Passing
Information via
Query Strings

3

Passing
Information via
the URL Path

4

Cookies

5

Serialization

6

Session State

7

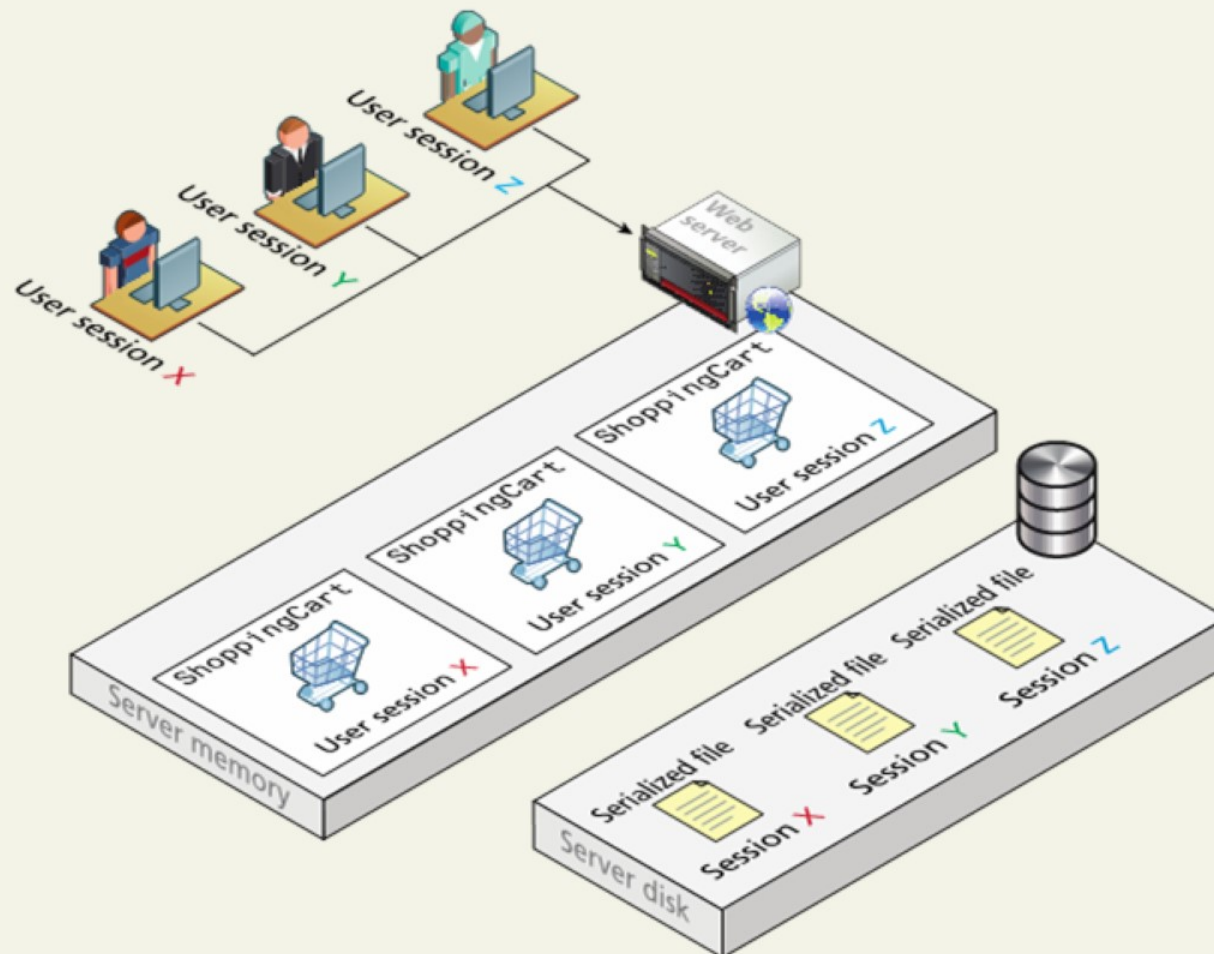
HTML5 Web
Storage

8

Caching

Session State

How Does Session State Work?



Session State

Accessing session state

```
<?php
session_start();
if ( isset($_SESSION['user']) ) {
    // User is logged in
}
else {
    // No one is logged in (guest)
}
?>
```

Session State

Checking session existence

```
<?php
include_once("ShoppingCart.class.php");

session_start();

// always check for existence of session object before accessing it
if ( !isset($_SESSION["Cart"]) ) {
    // session variables can be strings, arrays, or objects, but
    // smaller is better

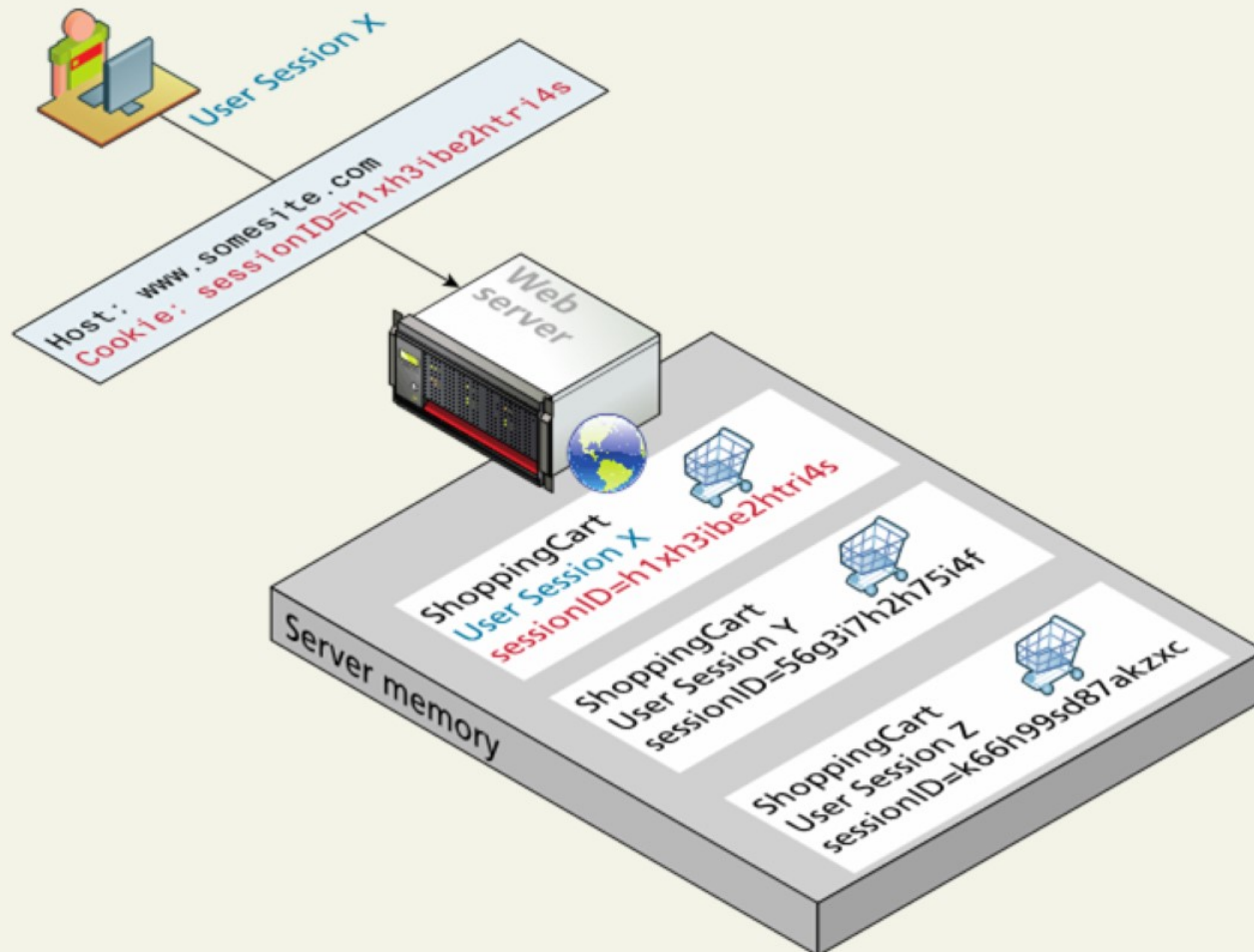
    $_SESSION["Cart"] = new ShoppingCart();
}

$cart = $_SESSION["Cart"];

?>
```

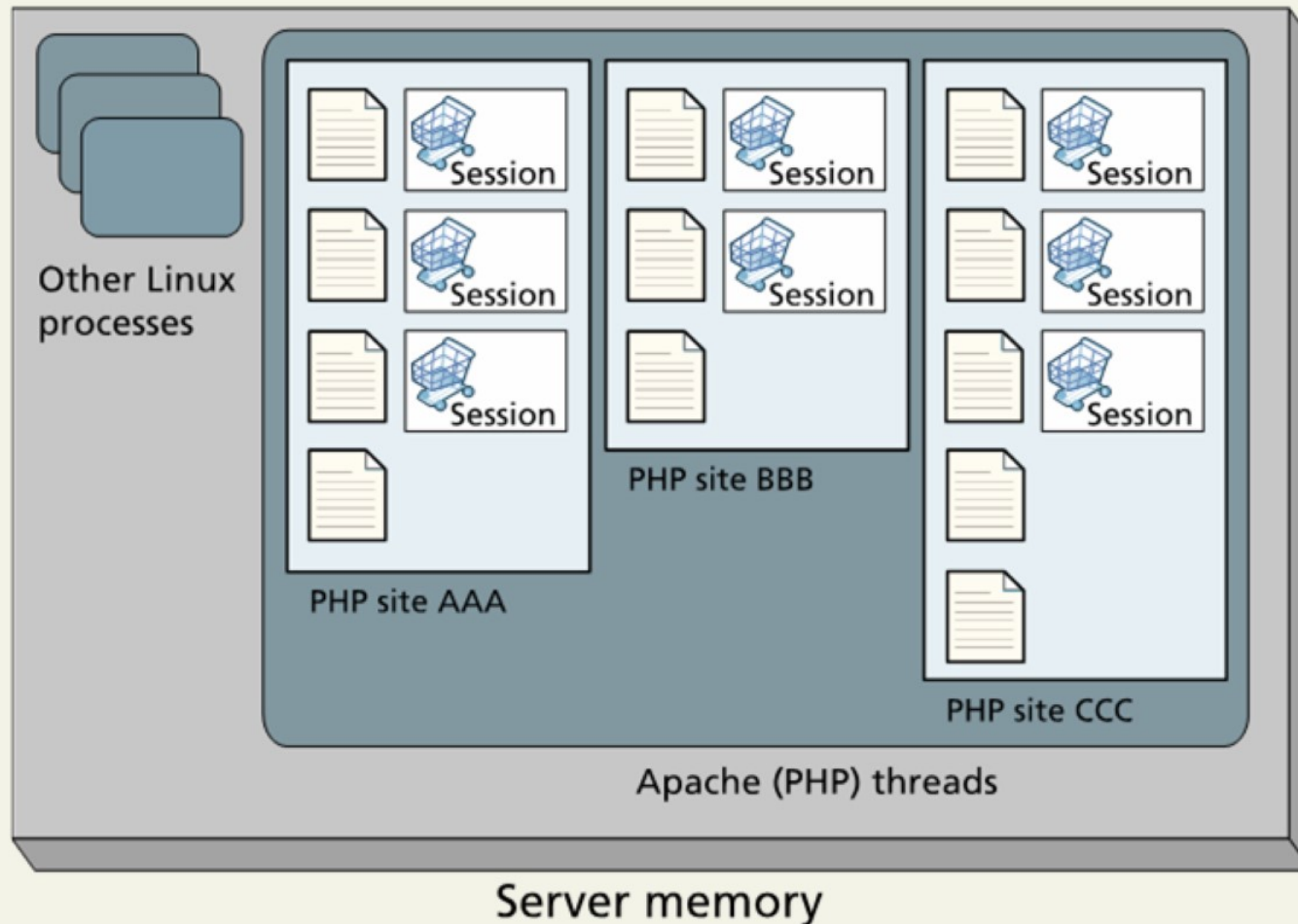

Session State

How does it all work



Session State

Session Storage and Configuration



Chapter 16

1

The Problem of
State in Web
Applications

2

Passing
Information via
Query Strings

3

Passing
Information via
the URL Path

4

Cookies

5

Serialization

6

Session State

7

HTML5 Web
Storage

8

Caching

HTML5 Web Storage

Using Web Storage - Writing

```
<form ... >
<h1>Web Storage Writer</h1>
<script language="javascript" type="text/javascript">
if (typeof (localStorage) === "undefined" ||
    typeof (sessionStorage) === "undefined") {
    alert("Web Storage is not supported on this browser...");
}
else {
    sessionStorage.setItem("TodaysDate", new Date());
    sessionStorage.FavoriteArtist = "Matisse";
    localStorage.UserName = "Ricardo";
    document.write("web storage modified");
}
</script>
<p><a href="WebStorageReader.php">Go to web storage reader</a></p>
</form>
```

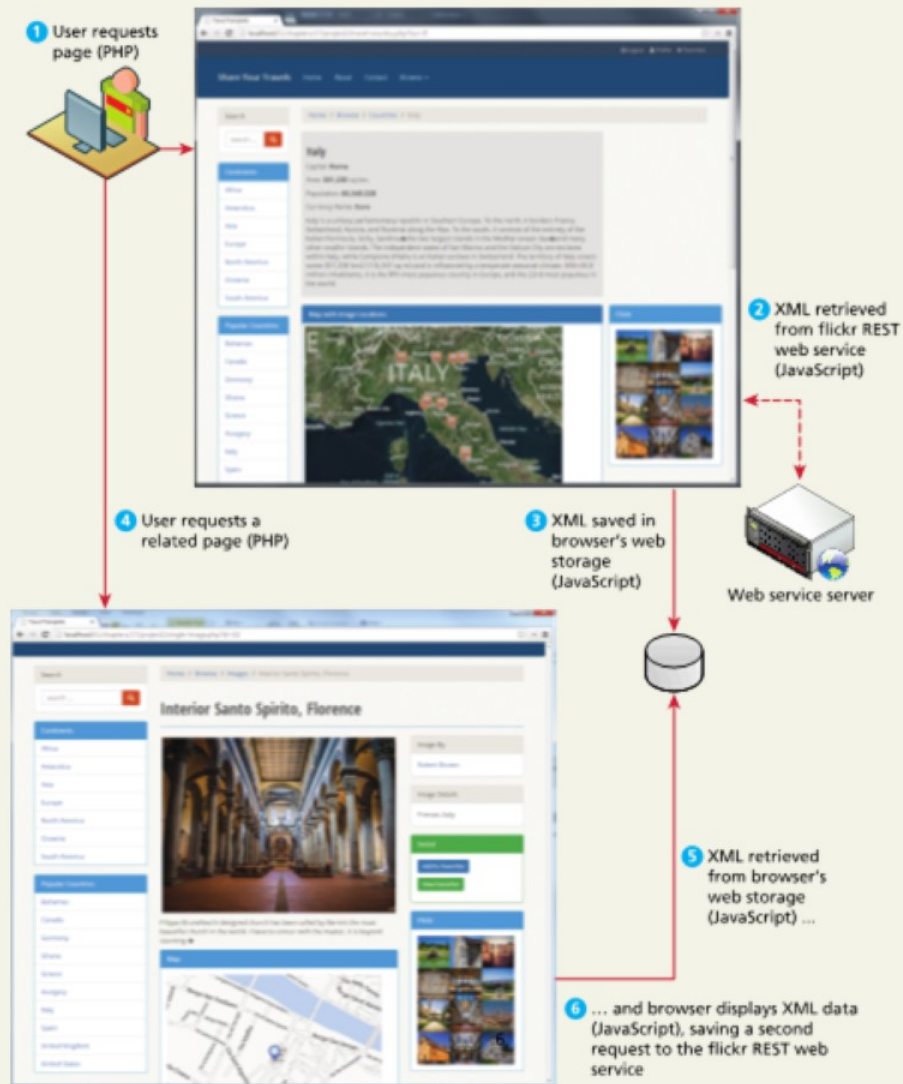
HTML5 Web Storage

Using Web Storage - Reading

```
<form id="form1" runat="server">
<h1>Web Storage Reader</h1>
<script language="javascript" type="text/javascript">
    if (typeof (localStorage) === "undefined" ||
        typeof (sessionStorage) === "undefined") {
        alert("Web Storage is not supported on this browser...");
    }
    else {
        var today = sessionStorage.getItem("TodaysDate");
        var artist = sessionStorage.FavoriteArtist;
        var user = localStorage.UserName;
        document.write("date saved=" + today);
        document.write("<br/>favorite artist=" + artist);
        document.write("<br/>user name = " + user);
    }
</script>
</form>
```

HTML5 Web Storage

Why Would We Use Web Storage?



Chapter 16

1

The Problem of
State in Web
Applications

2

Passing
Information via
Query Strings

3

Passing
Information via
the URL Path

4

Cookies

5

Serialization

6

Session State

7

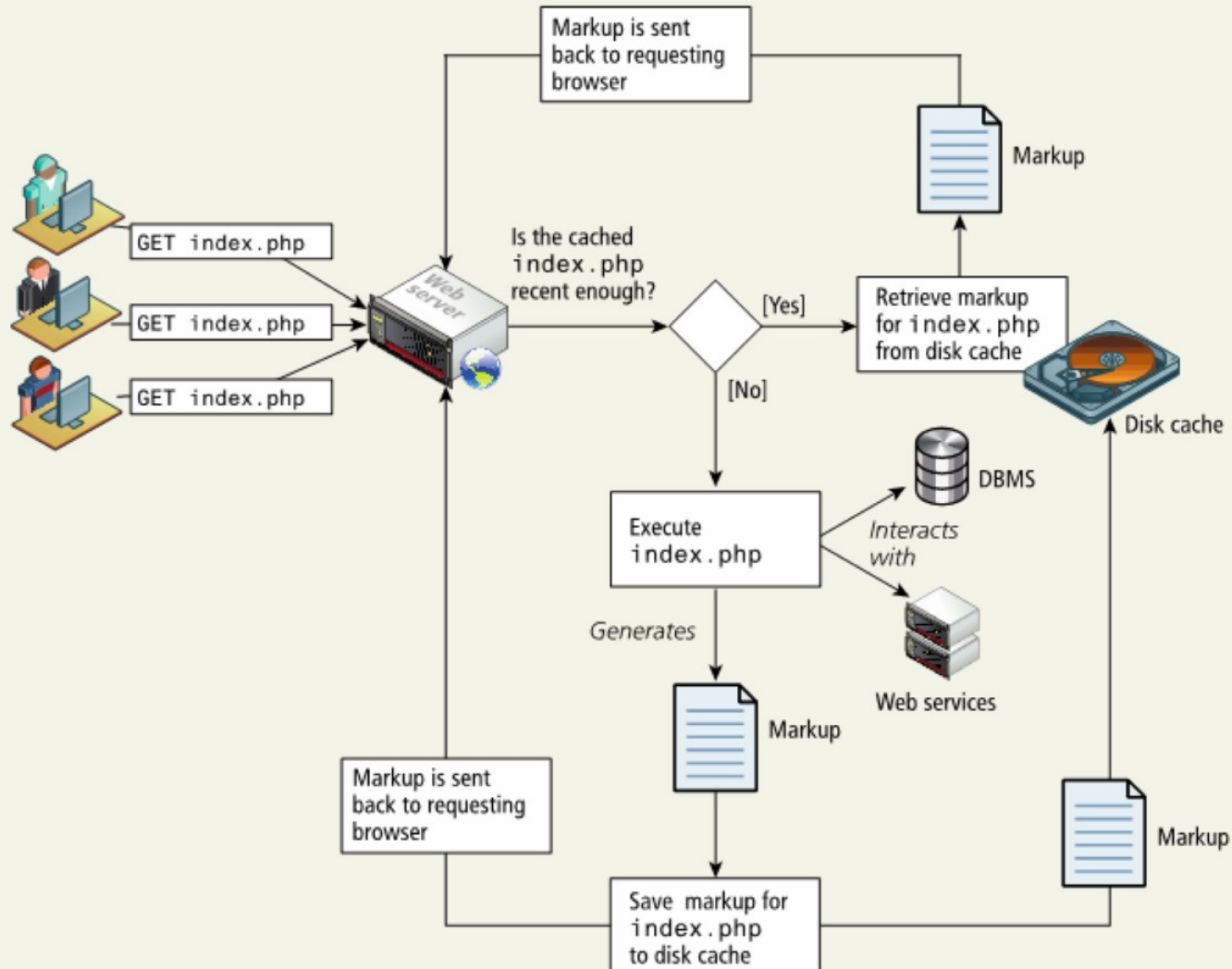
HTML5 Web
Storage

8

Caching

Caching

Page Output Caching



Caching

Application Data Caching

Not always prudent to cache entire pages (think customizations for logged in users)

Nice to programmatically place commonly used collections of data that require time-intensive queries from the database or web server into cache memory, and then other pages that also need that same data can use the cache version rather than re-retrieve it from its original location

Read about memcache.

Chapter 16 cont.

9

Summary

Summary

Key Terms

application page output session state
data caching caching
cache persistent
 cookies web storage
cookies
 serialization
HttpOnly
cookie session
 cookie

Summary

Questions?