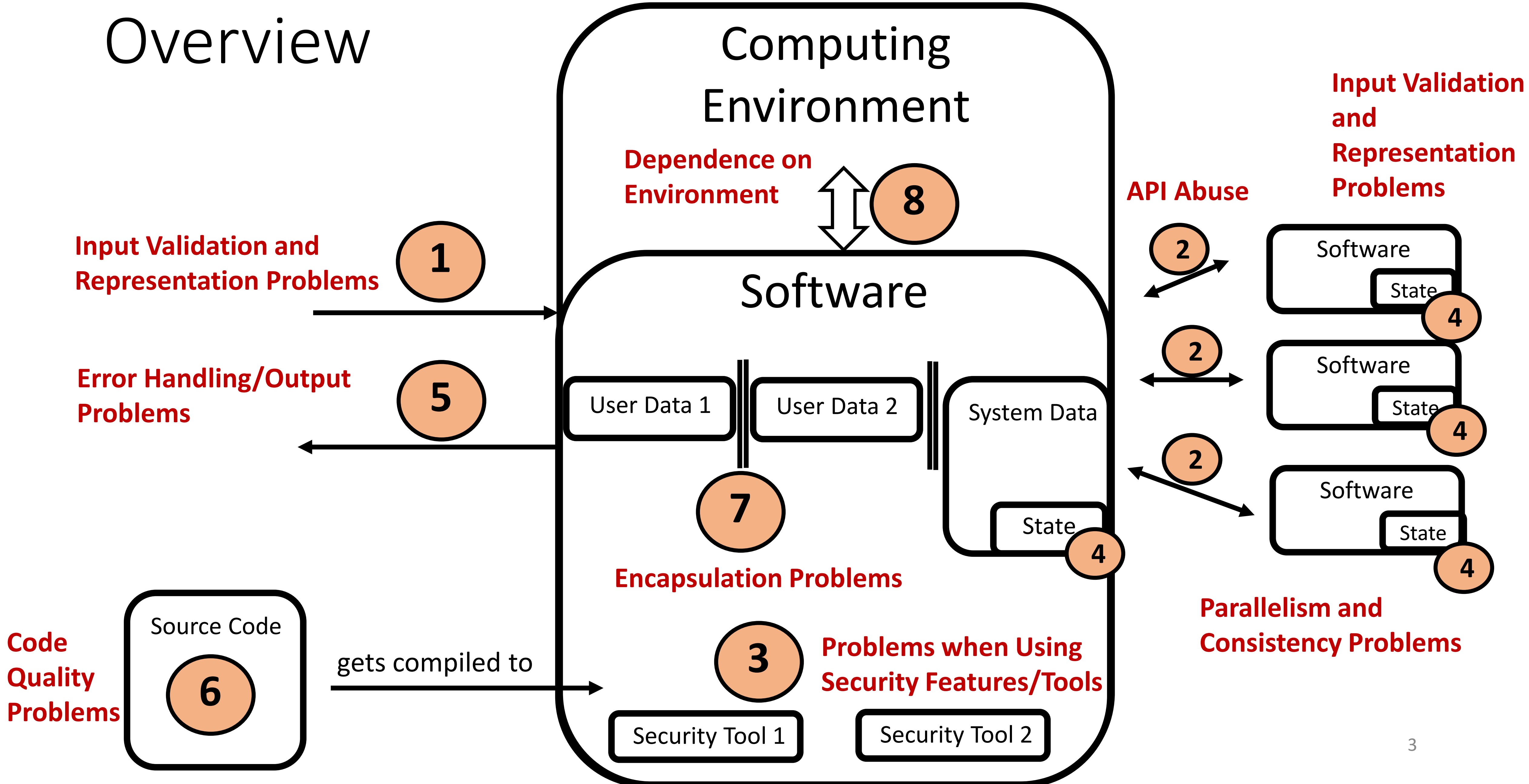
# Software Security 08

Sven Schäge

## Web-Security Introduction

## Overview



Computing Environment
Dependence on Environment
8
Input Validation and Representation Problems
1
Error Handling/Output Problems
5
Software
User Data 1
User Data 2
System Data
State
4
7
Encapsulation Problems
3
Problems when Using Security Features/Tools
Security Tool 1
Security Tool 2
Source Code
6
Code Quality Problems
gets compiled to
API Abuse
2
Input Validation and Representation Problems
Software
State
4
Parallelism and Consistency Problems

## The CWE Top 25

Below is a list of the weaknesses in the 2022 CWE Top 25, including the overall score of each. The KEV Count (CVEs) shows the number of CVE-2020/CVE-2021 Records from the CISA KEV list that were mapped to the given weakness.

| Rank | ID | Name | Score | KEV Count (CVEs) | Rank Change vs. 2021 |
|---|---|---|---|---|---|
| 1 | CWE-787 | Out-of-bounds Write | 64.20 | 62 | 0 |
| 2 | CWE-79 | Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') | 45.97 | 2 | 0 |
| 3 | CWE-89 | Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') | 22.11 | 7 | +3 ▲ |
| 4 | CWE-20 | Improper Input Validation | 20.63 | 20 | 0 |
| 5 | CWE-125 | Out-of-bounds Read | 17.67 | 1 | -2 ▼ |
| 6 | CWE-78 | Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') | 17.53 | 32 | -1 ▼ |
| 7 | CWE-416 | Use After Free | 15.50 | 28 | 0 |
| 8 | CWE-22 | Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') | 14.08 | 19 | 0 |
| 9 | CWE-352 | Cross-Site Request Forgery (CSRF) | 11.53 | 1 | 0 |
| 10 | CWE-434 | Unrestricted Upload of File with Dangerous Type | 9.56 | 6 | 0 |
| 11 | CWE-476 | NULL Pointer Dereference | 7.15 | 0 | +4 ▲ |
| 12 | CWE-502 | Deserialization of Untrusted Data | 6.68 | 7 | +1 ▲ |
| 13 | CWE-190 | Integer Overflow or Wraparound | 6.53 | 2 | -1 ▼ |
| 14 | CWE-287 | Improper Authentication | 6.35 | 4 | 0 |
| 15 | CWE-798 | Use of Hard-coded Credentials | 5.66 | 0 | +1 ▲ |
| 16 | CWE-862 | Missing Authorization | 5.53 | 1 | +2 ▲ |
| 17 | CWE-77 | Improper Neutralization of Special Elements used in a Command ('Command Injection') | 5.42 | 5 | +8 ▲ |
| 18 | CWE-306 | Missing Authentication for Critical Function | 5.15 | 6 | -7 ▼ |
| 19 | CWE-119 | Improper Restriction of Operations within the Bounds of a Memory Buffer | 4.85 | 6 | -2 ▼ |
| 20 | CWE-276 | Incorrect Default Permissions | 4.84 | 0 | -1 ▼ |
| 21 | CWE-918 | Server-Side Request Forgery (SSRF) | 4.27 | 8 | +3 ▲ |
| 22 | CWE-362 | Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition') | 3.57 | 6 | +11 ▲ |
| 23 | CWE-400 | Uncontrolled Resource Consumption | 3.56 | 2 | +4 ▲ |
| 24 | CWE-611 | Improper Restriction of XML External Entity Reference | 3.38 | 0 | -1 ▼ |
| 25 | CWE-94 | Improper Control of Generation of Code ('Code Injection') | 3.32 | 4 | +3 ▲ |

# Websurfing

## Overview

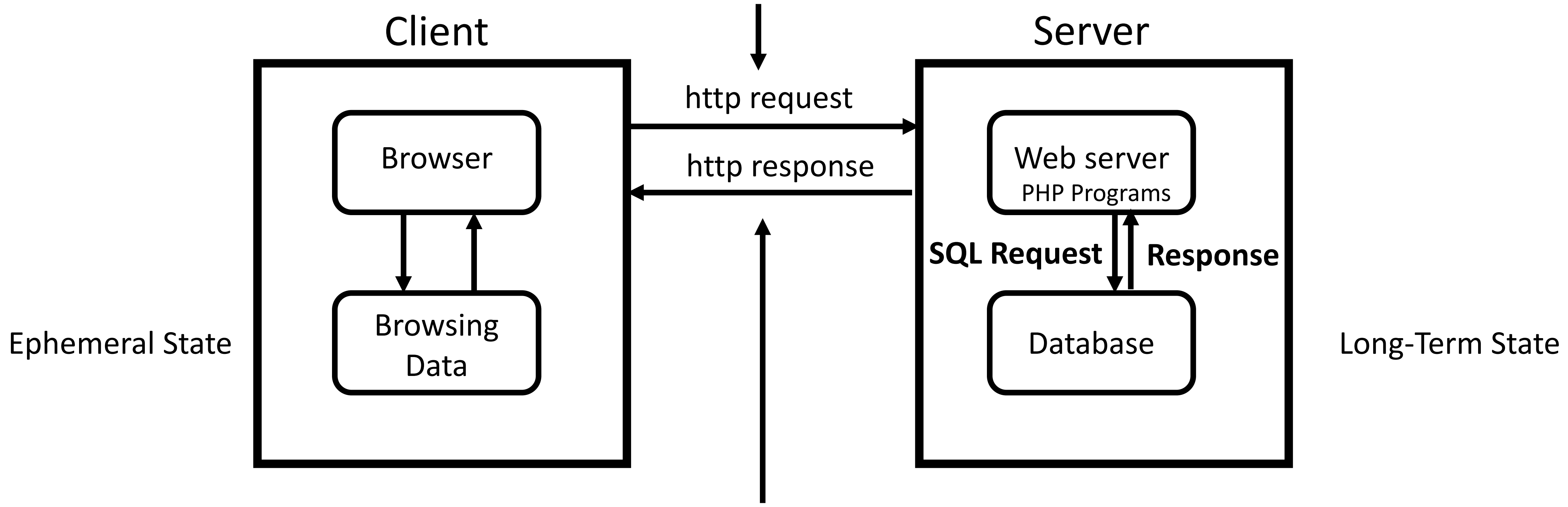
- The Web, http, and Databases
- SQL injection

- The Web and Ephemeral States
- Session Hijacking
- Cross-Site Request Forgery (CSRF)

- The Web and Mobile Code
- Cross-Site Scripting

## Websurfing

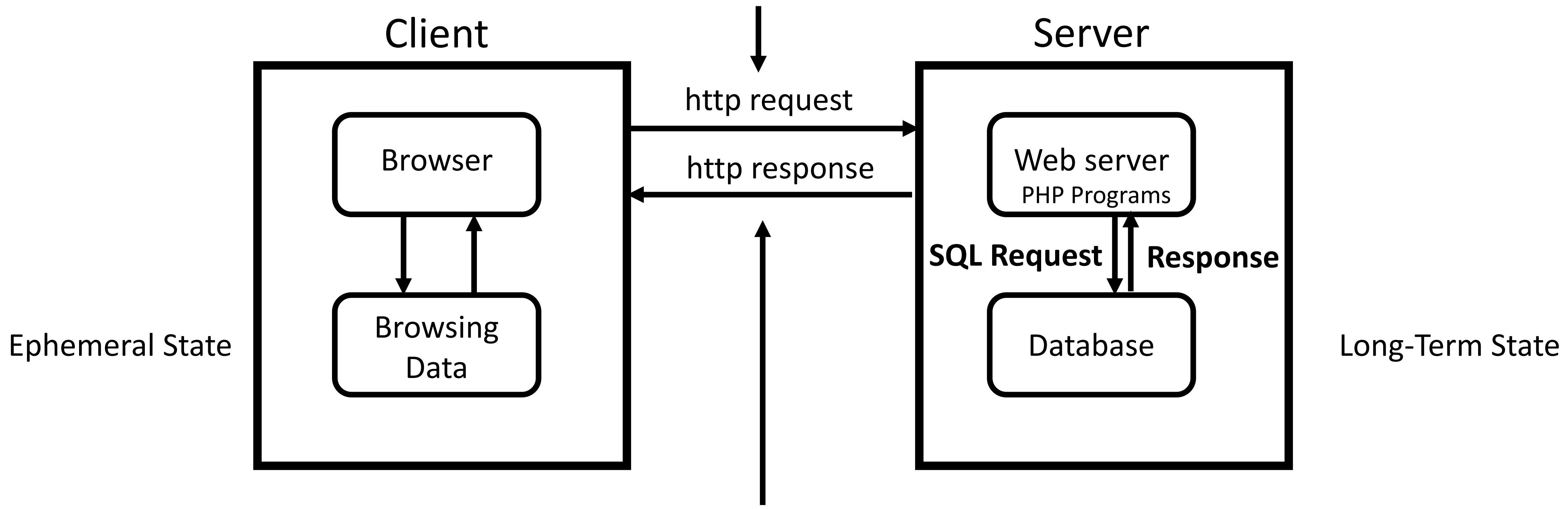
Browser essentially sends URL in format:
protocol://ServerAddress/filePathOnServer?Argument1&Argument2&Argument3…
e.g. https://en.wikipedia.org/w/index.php?title=Hypertext_Transfer_Protocol&action=edit&section=10



http response may contain static (usually .html) files or dynamically generated files. The latter are usually output by a PHP program that runs on the server and communicates with the database

# Websurfing

Two request types usually used for websurfing:
GET: URL contains all information required to process request, request to read-only
POST: may change data, input data in explicit data fields, e.g. posting to forum

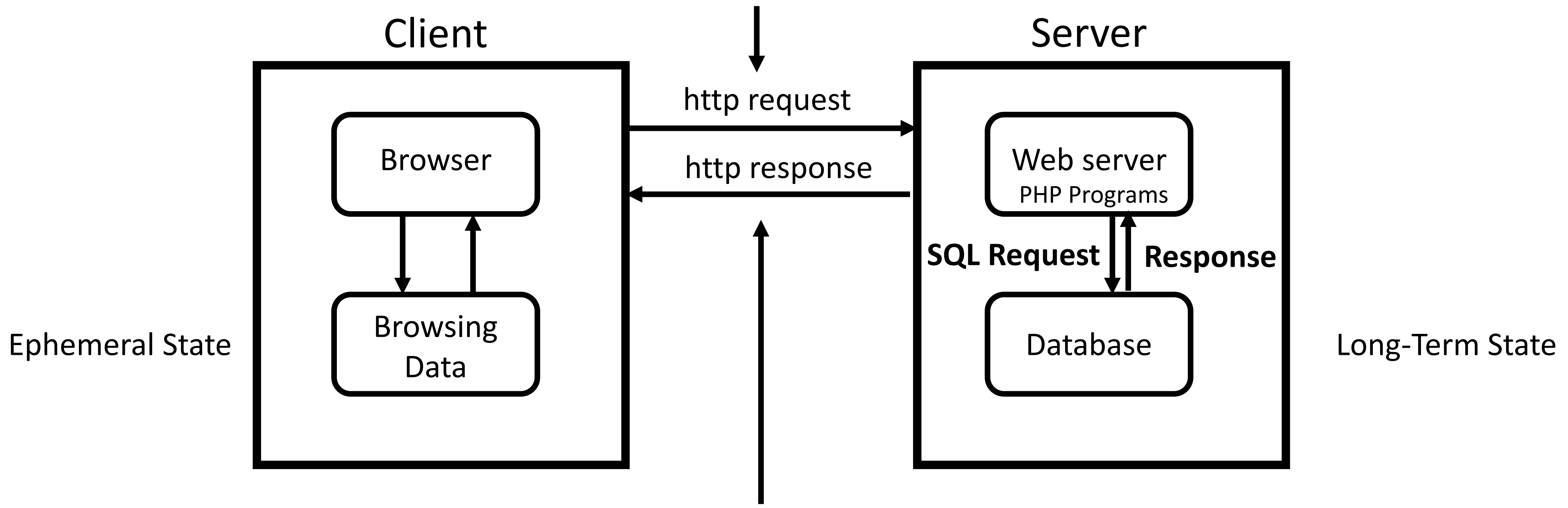


Response will contain cookies that should be stored on client to maintain some ephemeral state

## SQL

# SQL and Websurfing

Browser essentially sends URL in format:
protocol://ServerAddress/filePathOnServer?Argument1&Argument2&Argument3…
e.g. https://en.wikipedia.org/w/index.php?title=Hypertext_Transfer_Protocol&action=edit&section=10



http response may contain static (usually .html) files or dynamically generated files. The latter are usually output by a PHP program that runs on the server and communicates with the database

## Why Database Management Systems (DBMS)?

- Help to keep long-term data consistent and valid!

- DBMSs implement ACID transactions that guarantee validity of data even in the face of problems like errors, power failures
- Atomic
  - Statements treated as a single unit, either all goes through or nothing at all
- Consistent
  - Database moves only from one valid state to another
  - Temporary invalid states not written to database, written data must adhere to all rules
- Isolation
  - Concurrently executed transactions behave as if done sequentially
  - Concurrency control
- Durability
  - Once a transcation finishes it will remain persistent, e.g. even after power failure

# SQL Language in Examples

Table Accounts

| UserID | Password | Balance |
|---|---|---|
| Alice | 1234 | 100 |
| Bob | 5678 | 200 |

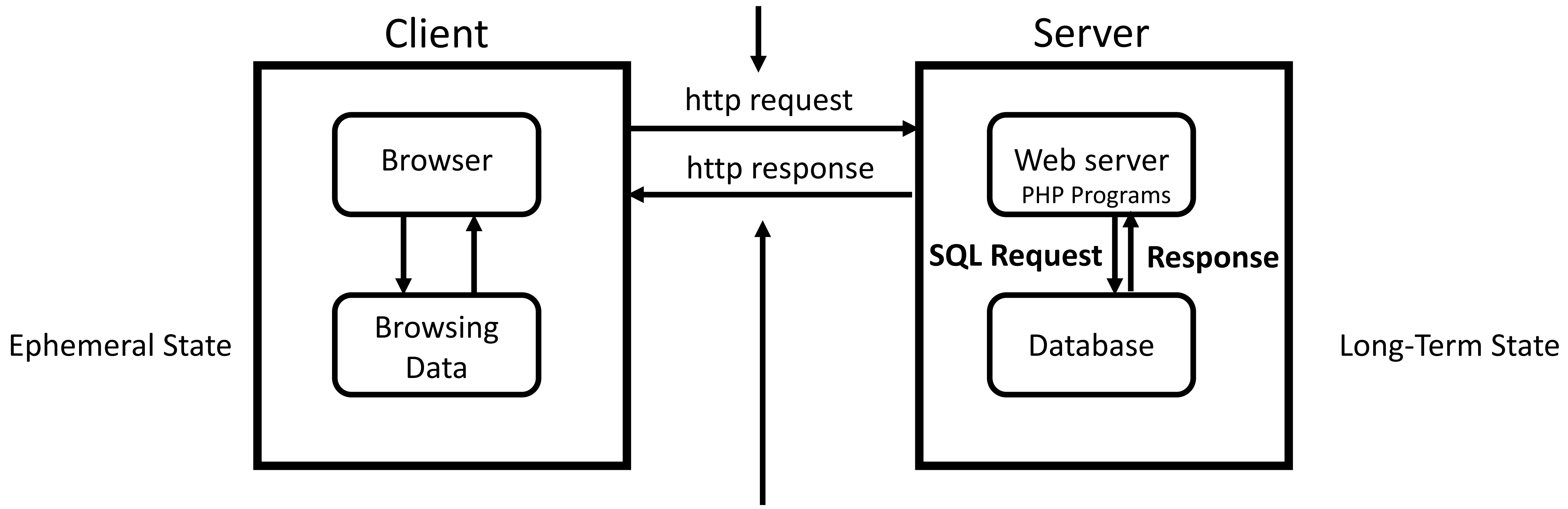
- Data stored in tables

- SELECT Balance FROM Accounts WHERE (UserID=‘Bob’ AND Password=‘5678’);
  - Reads Bob’s balance
- INSERT INTO Accounts Values(‘Charlie’, ‘90AB’, 200); -- A comment!
  - Adds new user Charlie together with his data
- UPDATE Accounts SET Balance=‘200’ WHERE UserID=‘Alice’;
  - Overwrites Alice’s balance to 200
- DROP TABLE Accounts; /*Yet another, possibly multiline comment*/
  - Deletes entire table

## PHP

## SQL and Websurfing

Browser essentially sends URL in format:
protocol://ServerAddress/filePathOnServer?Argument1&Argument2&Argument3…
e.g. https://en.wikipedia.org/w/index.php?title=Hypertext_Transfer_Protocol&action=edit&section=10



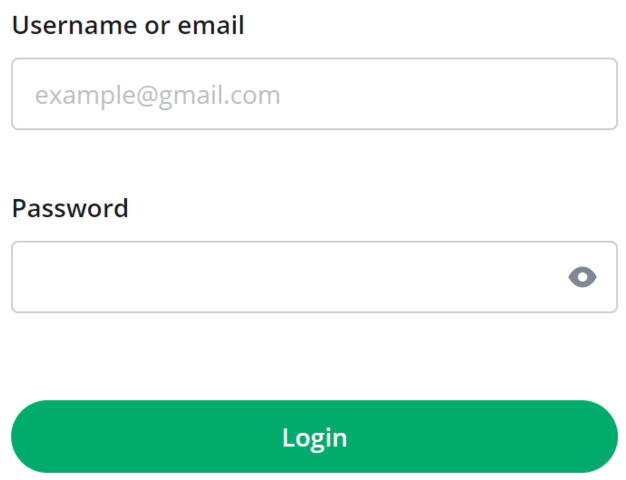
**http response may contain dynamically generated files that are usually output by a PHP program that runs on the server and communicates with the database**

# PHP Example

Table Accounts in Database

| UserID | Password | Balance |
|---|---|---|
| Alice | 1234 | 100 |
| Bob | 5678 | 200 |

Website presented to User



```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);
…
$valid=mysql_query(“SELECT * from Accounts WHERE (UserID=‘$usernameField’ AND Password=‘$passwordField’);”);
…
$conn->close();
?>

<!DOCTYPE html>
…
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
<input type="text" name="username" class="form-control" value="<?php echo $usernameField; ?>">
…
</html>
```