Back end Development

CS 4640
Programming Languages
for Web Applications

How User Interact With Web Apps

http://www.virginia.edu/

Usability

Web client

HTML
JS
CSS

HTTP
Request

HTTP
Response

Web application

Request

Response

PHPs

Web component$_1$

Web component$_2$

Web component$_n$

Web server

DB
or
persistent
data

DB server

Front end
(Client-side programming)

HTML, CSS, JS, UI/UX design

Back end
(Server-side programming)

Programming languages (such as Java, Python, PHP, Perl, Ruby), DBMS,
(production) web server, (local) development environment, web services,
functional and non-functional testing, architectures, algorithms, security, …

How the Web Software Works

PHP: Form Handling

Name:

> Awesome Dev

Email:

> awesome@virginia.edu

Comment:

> PHP is fun!

> Submit

Browser

```html
<form action="form-handler.php" method="post">
  <label>Name: </label>
  <input type="text" name="name" autofocus /> <br/>
  <label>Email:</label>
  <input type="email" name="emailaddr" /> <br/>
  <label>Comment: </label>
  <textarea rows="5" cols="40" name="comment"></textarea> <br/>

  <input type="submit" value="Submit" />
</form>
```

To server

```php
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST')
{
    # param => value
    $name = $_POST['name'];
    $email = $_POST['emailaddr'];
    $comment = $_POST['comment'];

    echo "Thanks for this comment, $name <br/>";
    echo "<i>$comment</i> <br/>";
    echo "We will reply to $email <br/>";
}
?>
```
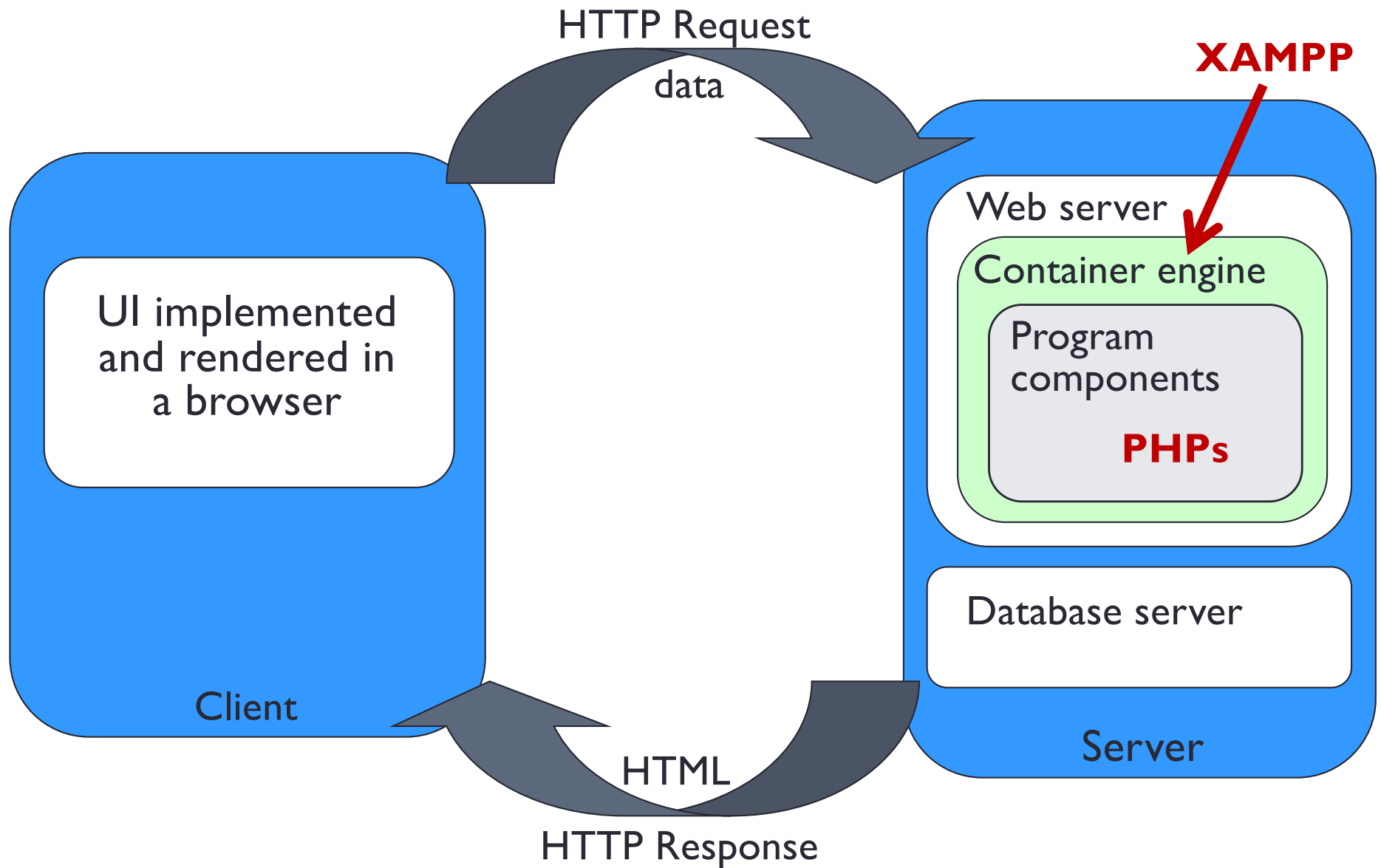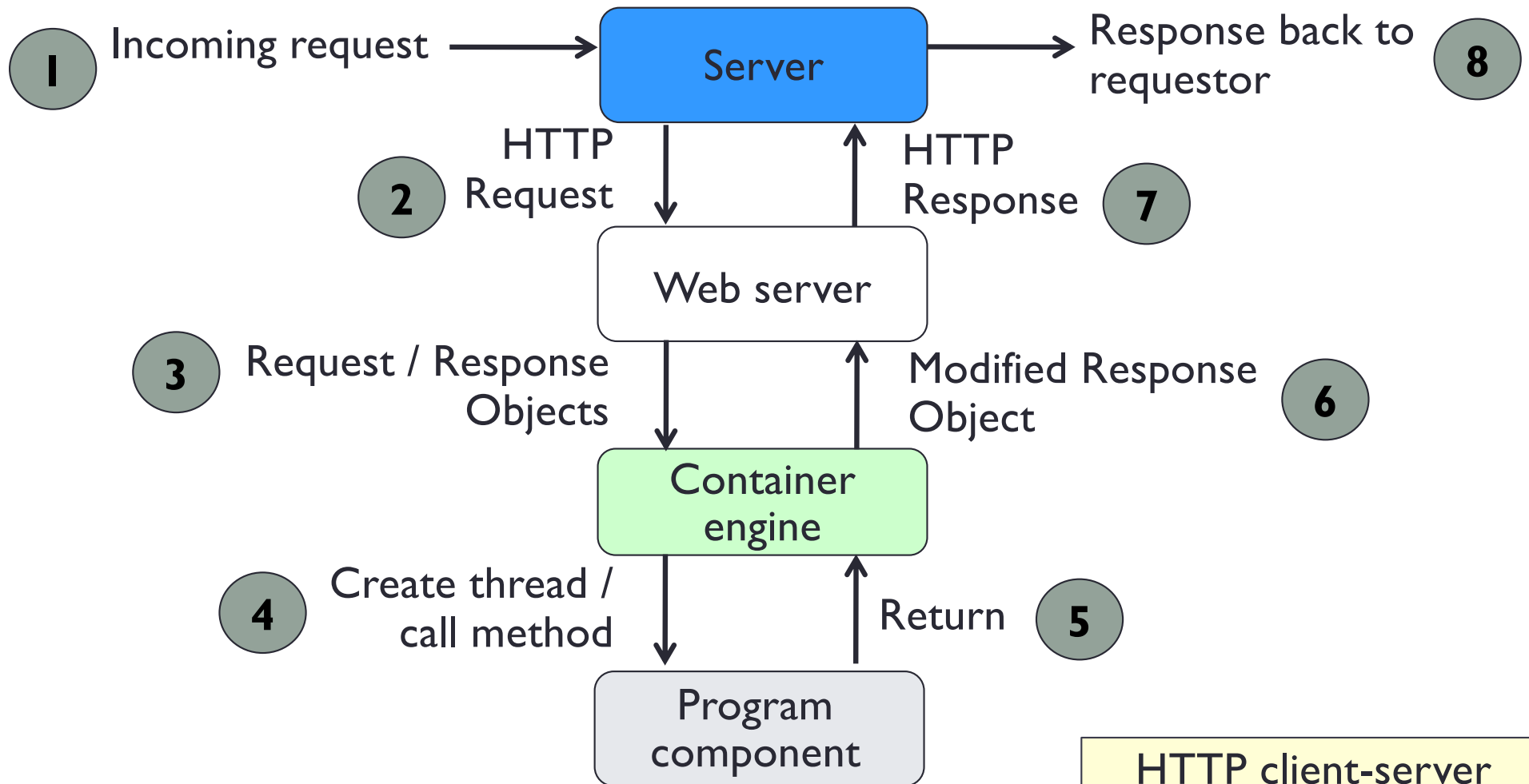
form-handler.php

To client

Thanks for this comment, Awesome Dev
PHP is fun!
We will reply to awesome@virginia.edu

Browser

Execution Overview

Session Management

How can servers keep track of state of different clients?

1. *Session*: A single coherent use of the system by the same user

   – Example : shopping carts

2. *Cookies*: A string of characters that a web server places on a browser's client to keep track of a session

   – Usually used as an index into a table (*dictionary*) on the server

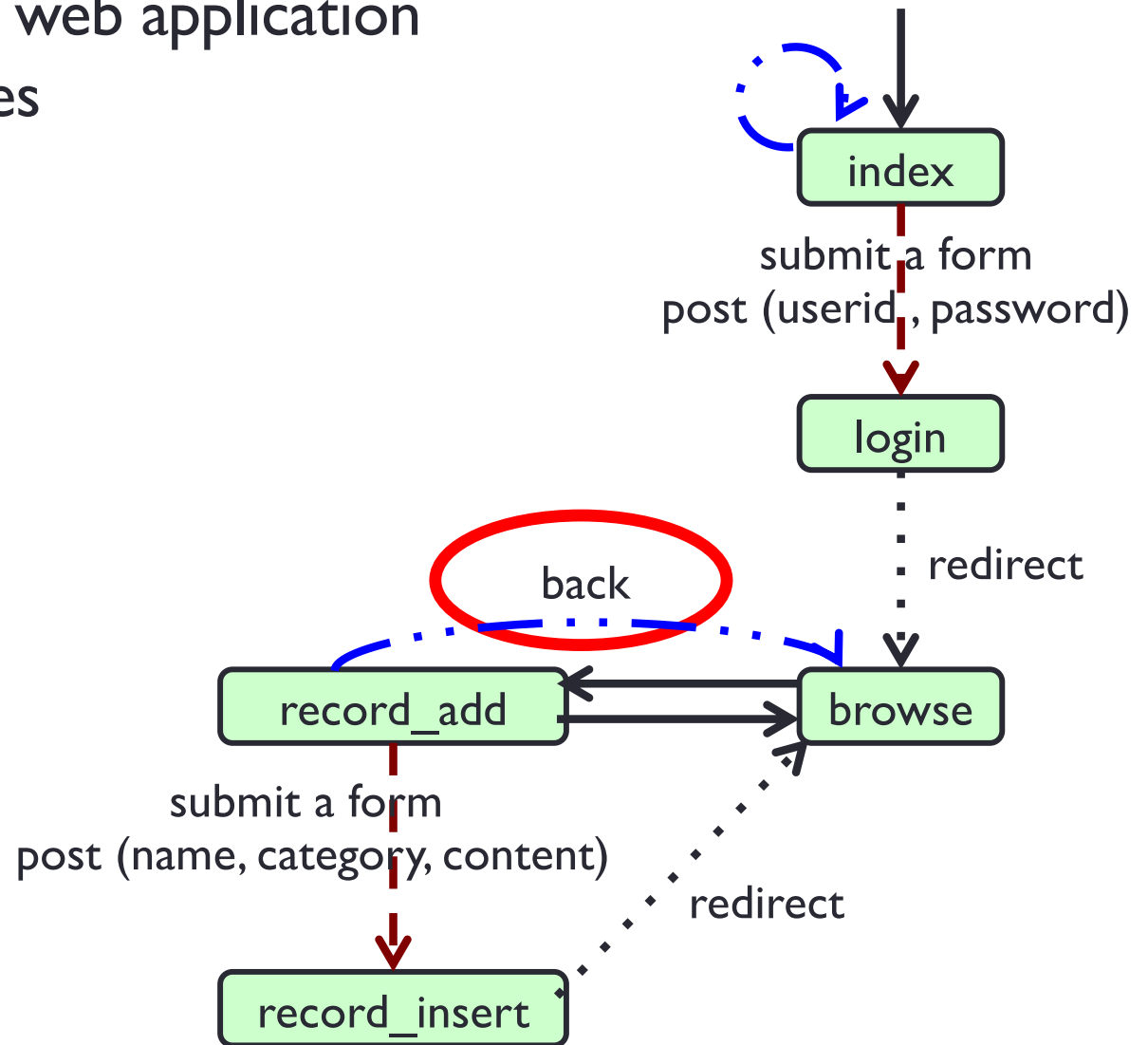   – Most dictionaries expire after a period of time (15 to 30 minutes)
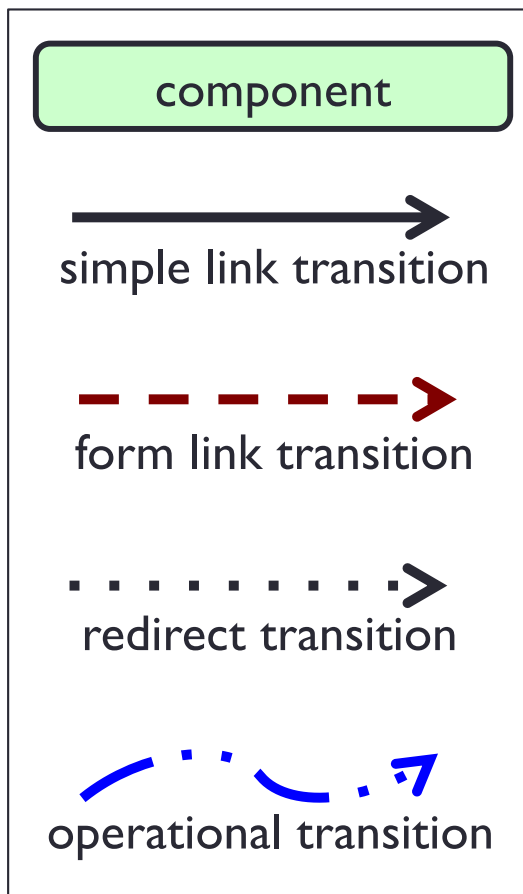
Additional mechanisms

- URL rewriting

- Hidden form control

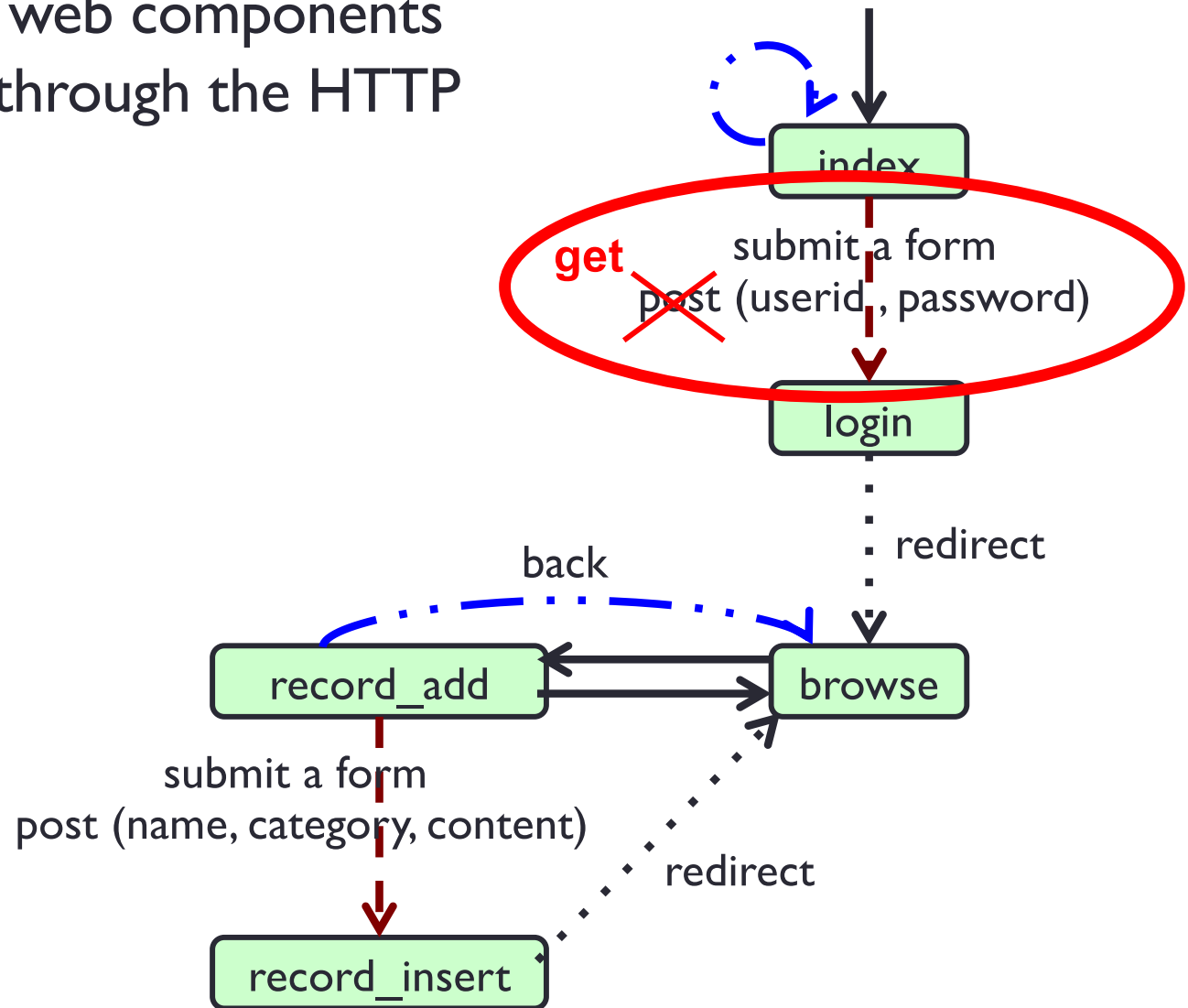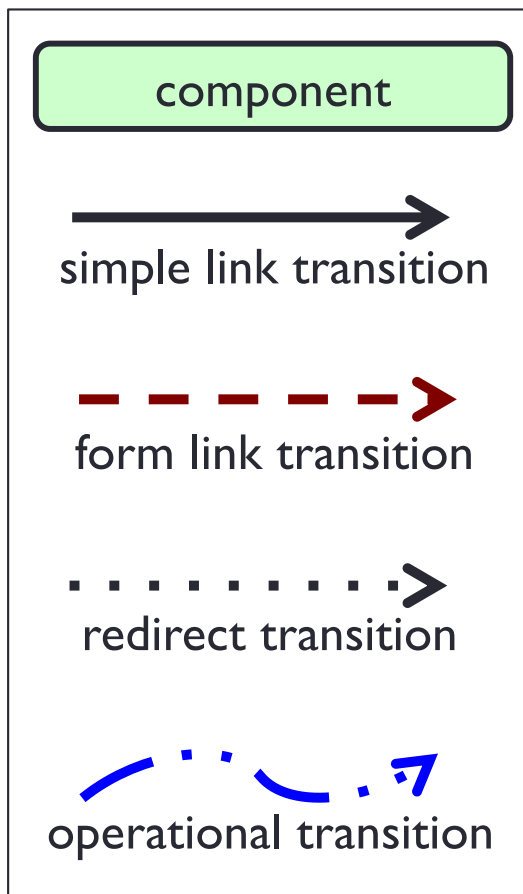We will come back to this later …

Additional Web Features

User's ability to control web application via web browser features

Additional Web Features (2)

Communication among web components depending on requests through the HTTP

Legend:
- **component** — simple link transition
- form link transition
- redirect transition
- operational transition

Diagram nodes: index → login (submit a form, ~~post~~ get (userid, password)) → browse (redirect) ; record_add ⇄ browse (back) ; record_add → record_insert (submit a form, post (name, category, content)) ; record_insert → browse (redirect)

Additional Web Features (3)

Control connections: forward, include, redirect

component

simple link transition

form link transition

redirect transition

operational transition

index

submit a form
post (userid, password)

login

forward

redirect

back

record_add

browse

submit a form
post (name, category, content)

redirect

record_insert