

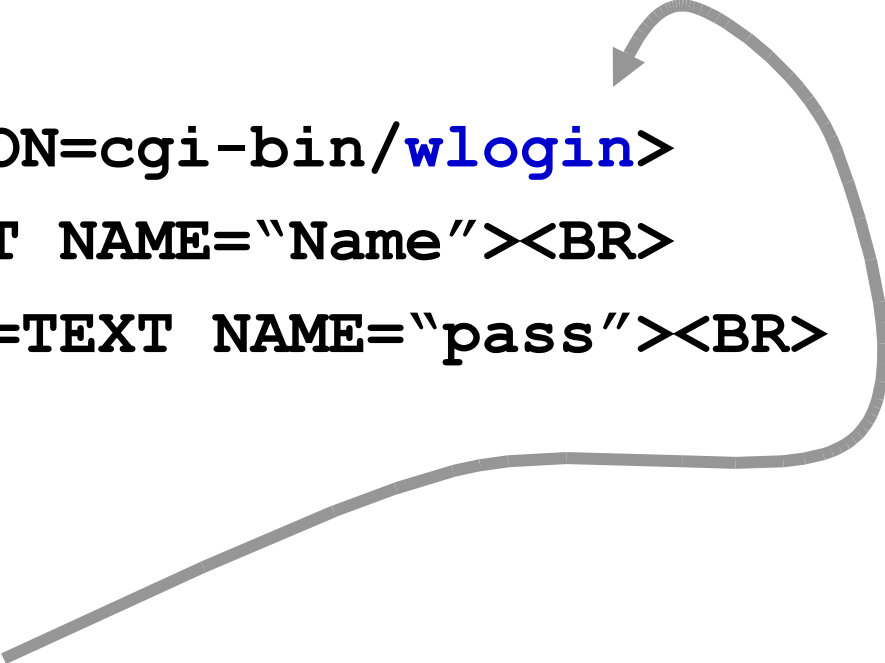
CGI, Forms & Web Applications

- A Web application typically consists of:
 - a number of HTML documents that include forms.
 - a number of CGI programs that receive *form submissions*.
- Sometimes a single CGI program can handle many different forms.

Simple Example: Restricted Access

- An initial web page contains a login form:

```
<FORM METHOD=GET ACTION=cgi-bin/wlogin>  
Name: <INPUT TYPE=TEXT NAME="Name"><BR>  
Password: <INPUT TYPE=TEXT NAME="pass"><BR>  
<INPUT TYPE=SUBMIT>  
</FORM>
```

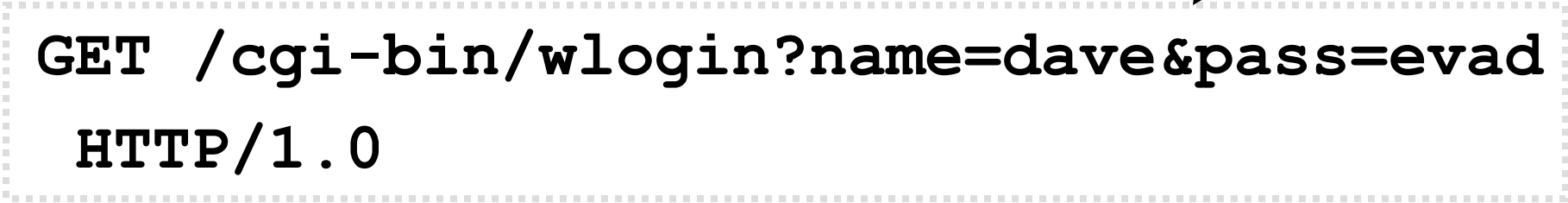


- A program named **wlogin** receives the name and password and decides what to send back.

Form Submission

- When the user presses the submit button, the browser sends a request:

this is all one line



```
GET /cgi-bin/wlogin?name=dave&pass=evad
HTTP/1.0
```

```
User-Agent: Netscape Navigator 4.7
```

What if POST instead of GET?

- IF the form METHOD=POST:

```
POST /cgi-bin/wlogin HTTP/1.0
```

```
Content-length: 19
```

```
name=dave&pass=evad
```

The **wlogin** program

- Get fields **name** and **pass**
 - the values entered by the user.
- Does something to decide if the name and password are valid.

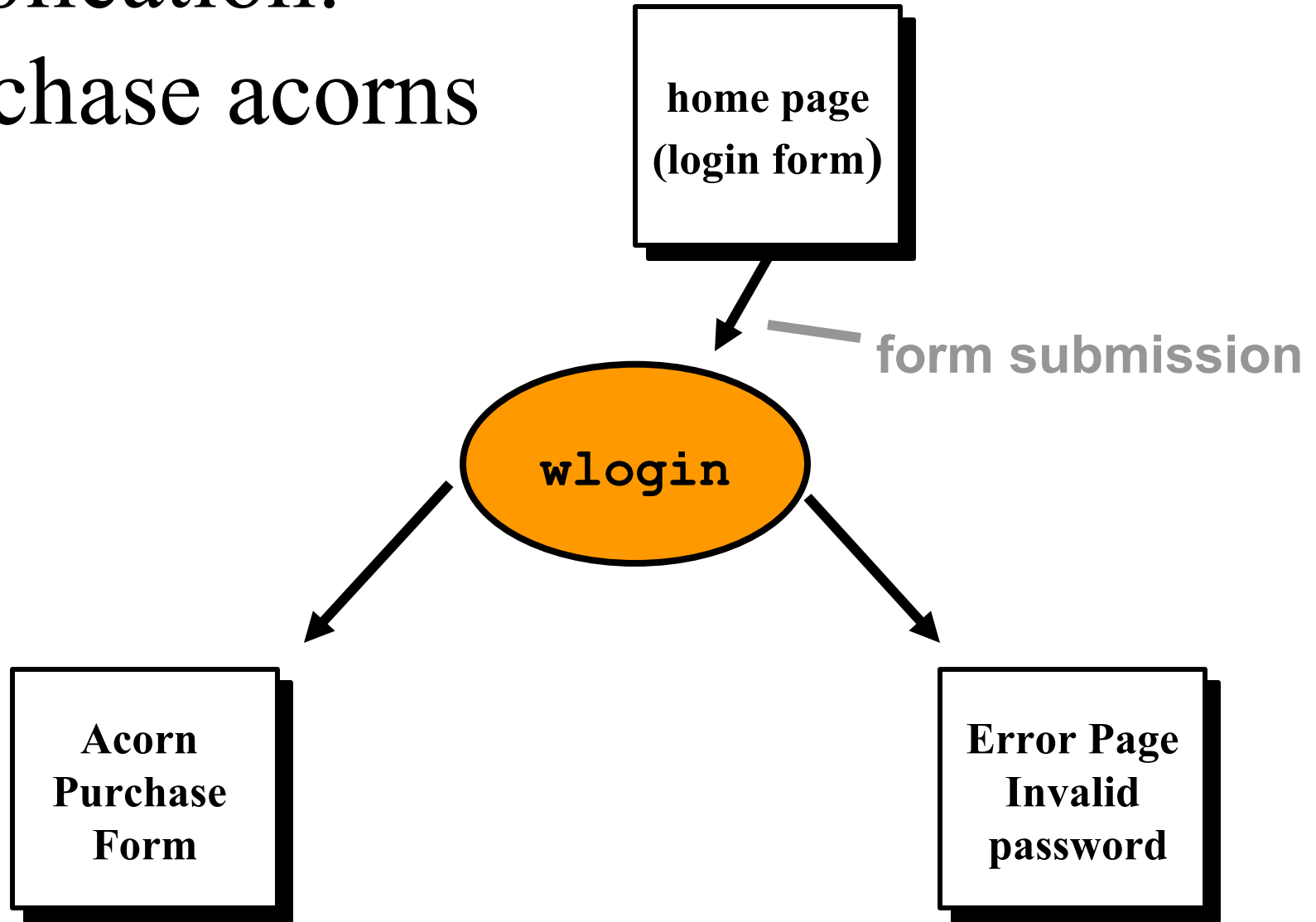
If valid:

send back some site Home Page (allow access)

Else

send back an error message (deny)

Application: purchase acorns



How many acorns do you want?

Acorn Purchase Form

```
<FORM METHOD=GET ACTION=cgi-bin/purch>
```

```
How many acorns do you want?
```

```
<INPUT TYPE=TEXT NAME="qty"><BR>
```

```
<INPUT TYPE=SUBMIT>
```

```
</FORM>
```

Query-line would look like:

```
GET /cgi-bin/purch?qty=64 HTTP/1.1
```

The **purch** program

- Get field **qty**
- Update a database:
 - how many acorns to ship
 - where to ship them?
 - who just bought them?
 - why do they want acorns?

purch knows this.
qty is part of the
request



purch has no
way of knowing !



The problem

- The HTTP request that was sent does not include information about who the user is!
- The request could come minutes, days or weeks after the login.
 - many other people could login in the meantime.

Sessions

- We need to establish some connection between a login form submission and an acorn form submission – to establish a *session*.
- We will look at a number of ways to support *sessions* later, for now we just need to understand the problem.

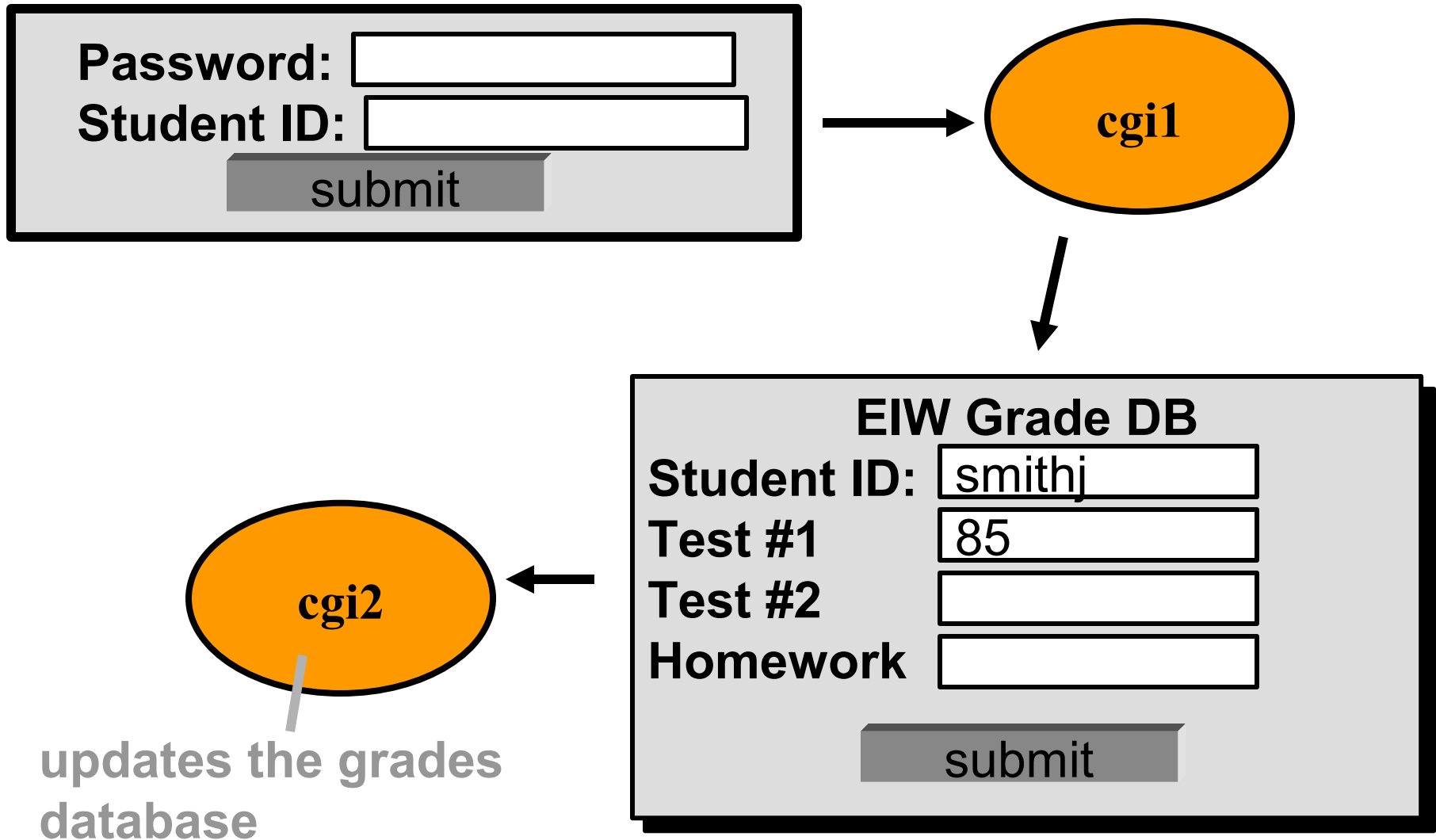
HTTP & CGI

- Each HTTP request is independent of others.
- We need to design systems so that many people can use the system at the same time.
- When using CGI – the external program gets the request and nothing else.
- The CGI program is started up each time a new request arrives.
 - The program is terminated when the response has been sent.

Another example

- Student Grades Database
- Initial Form:
 - Instructor Password
 - Student ID
- If the password is correct – a form is sent back that allows the instructor to enter grades for the student.

Student Grade Database



Hacking the Grades Database

- Notice that cgi2 does not require a password or instructor name!
- All we need to know is:
 - the names of the fields
 - the name of the program
- Test your skills:
<http://cgi2.cs.rpi.edu/~hollingd/websys/cde/StudentDB/>

The Real World

- We do actually make sure this can't happen!
- There have been some famous web systems that were developed improperly and were as “open” as the student grade database:
 - hotmail 1998 – anyone could read your mail.