Server-Side Web Programming: Python (Part 1)

Robert M. Dondero, Ph.D
Princeton University

Objectives

- We will cover:
  - Web application architectures
  - Python WSGI programming

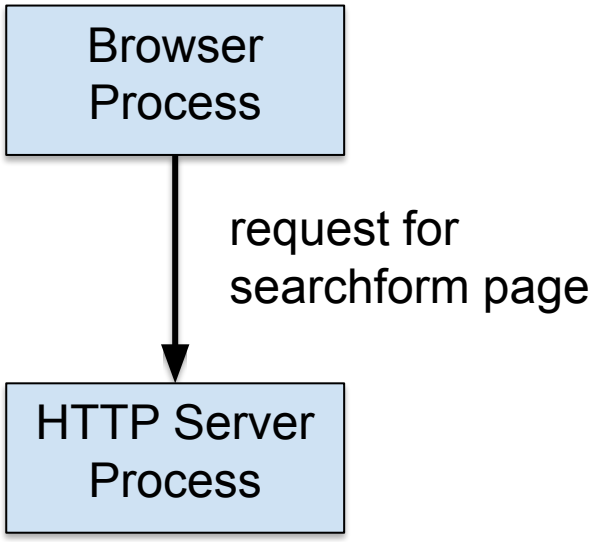
Agenda

- **Web application architectures**
- Python WSGI programming

Web App Architectures

- Consider Option 1…
  - Web app **forks a child process** to handle each HTTP request

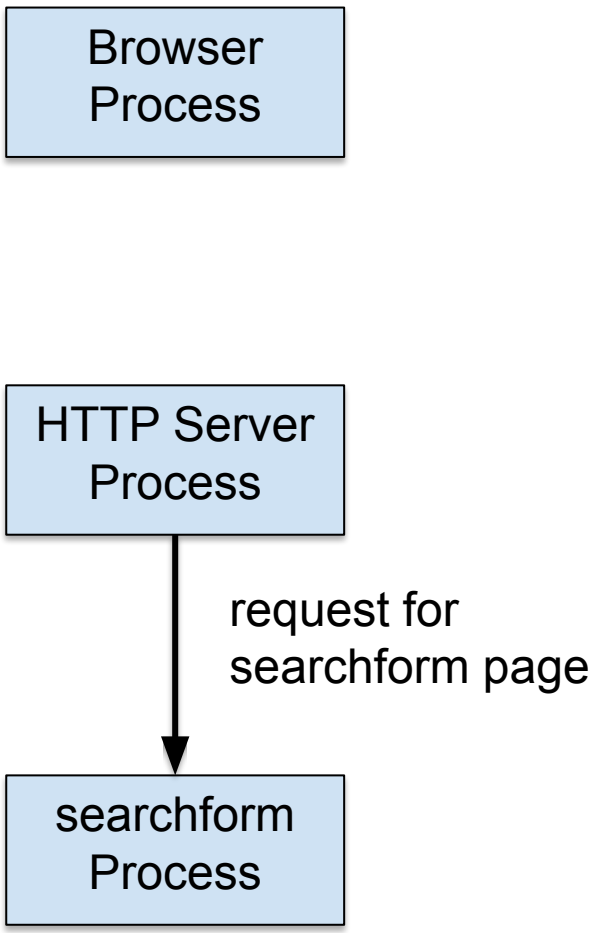
Web App Architectures



Option 1

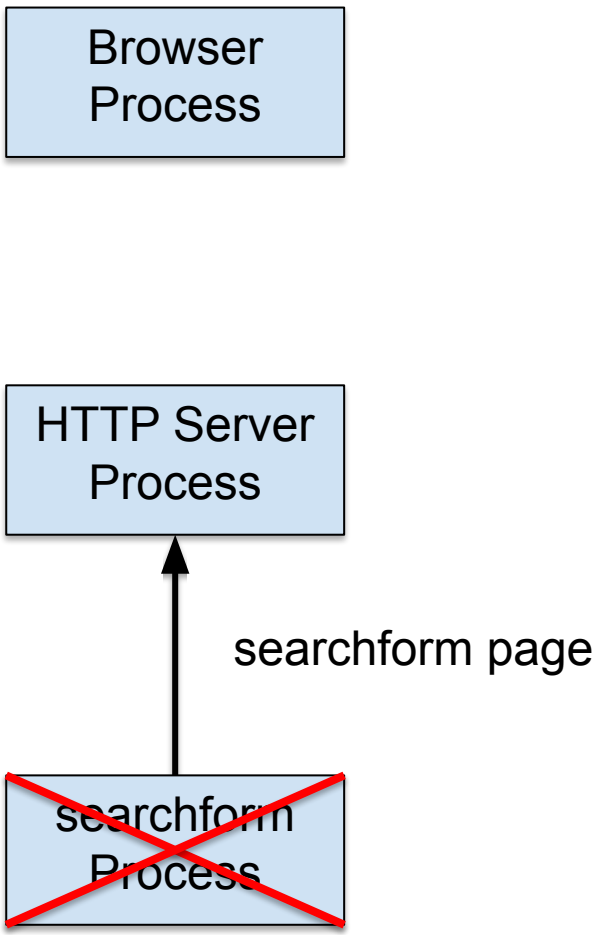
Web App Architectures

Option 1
(cont.)

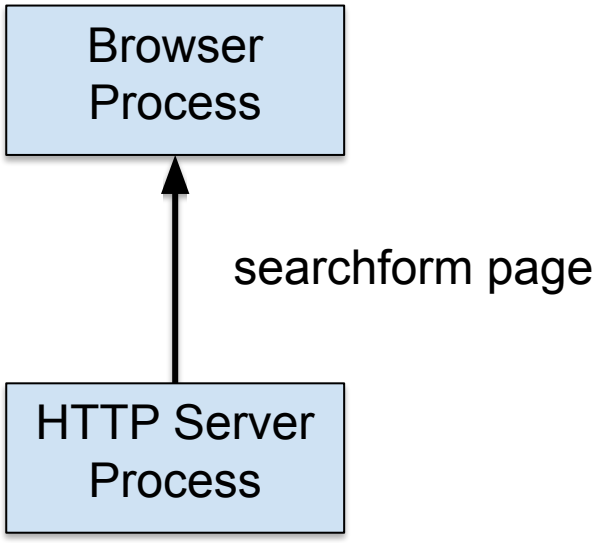


Web App Architectures

Option 1
(cont.)

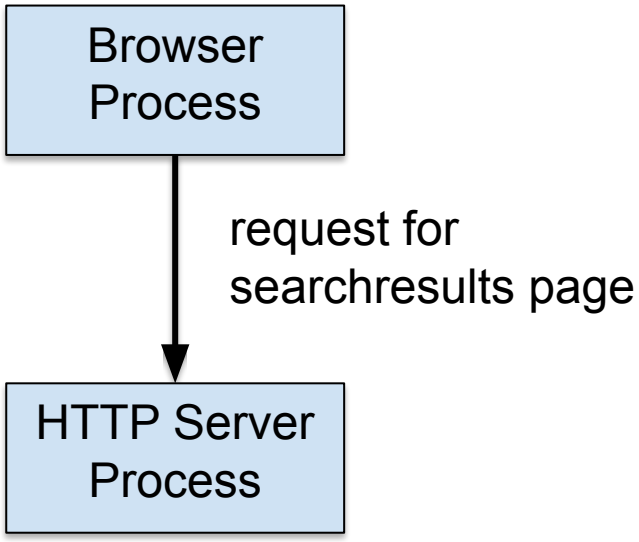


Web App Architectures



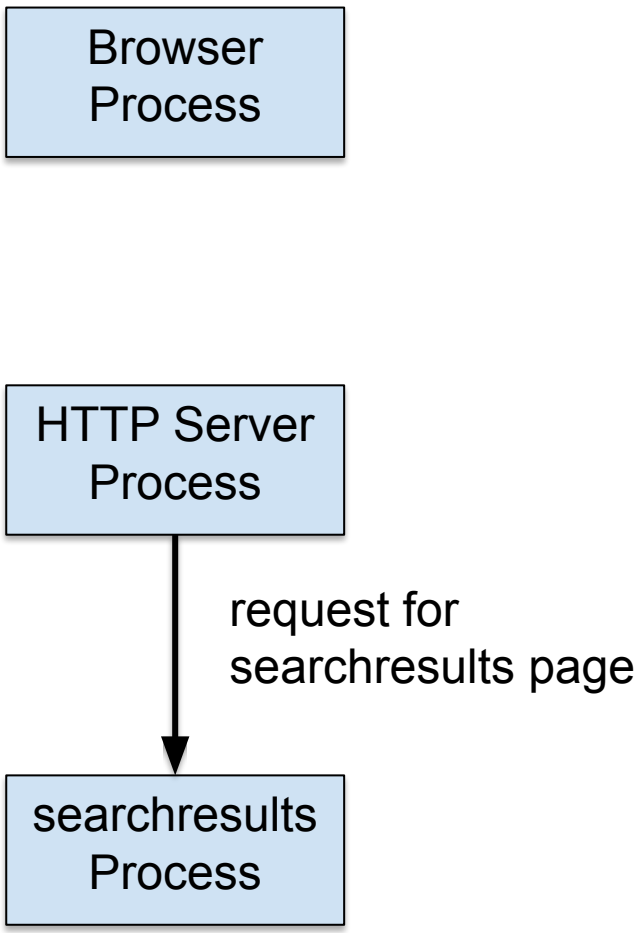
Option 1 (cont.)

Web App Architectures



Option 1 (cont.)

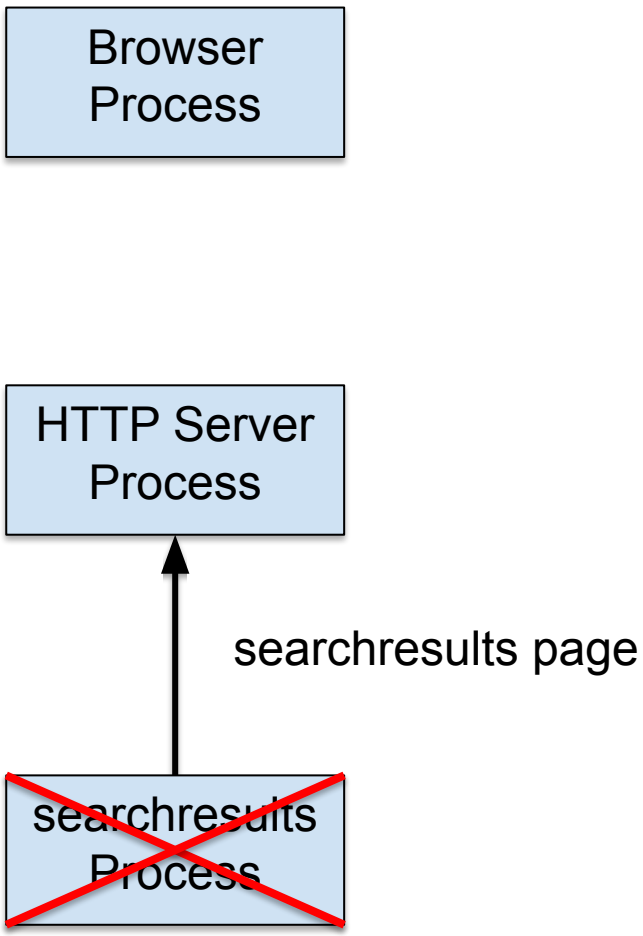
Web App Architectures



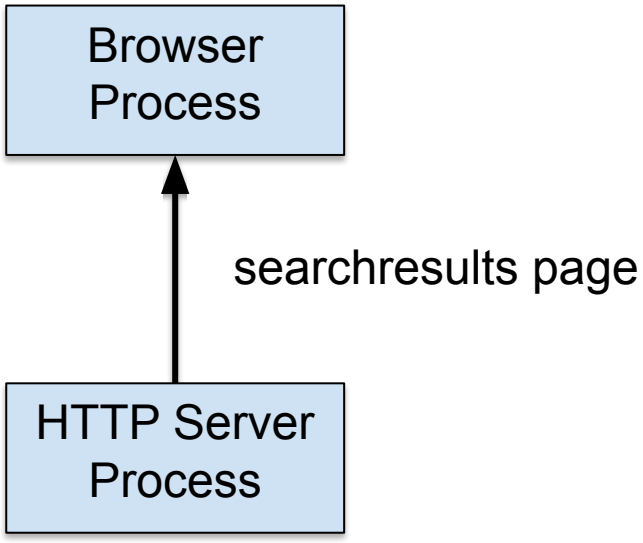
Option 1 (cont.)

Web App Architectures

Option 1
(cont.)



Web App Architectures



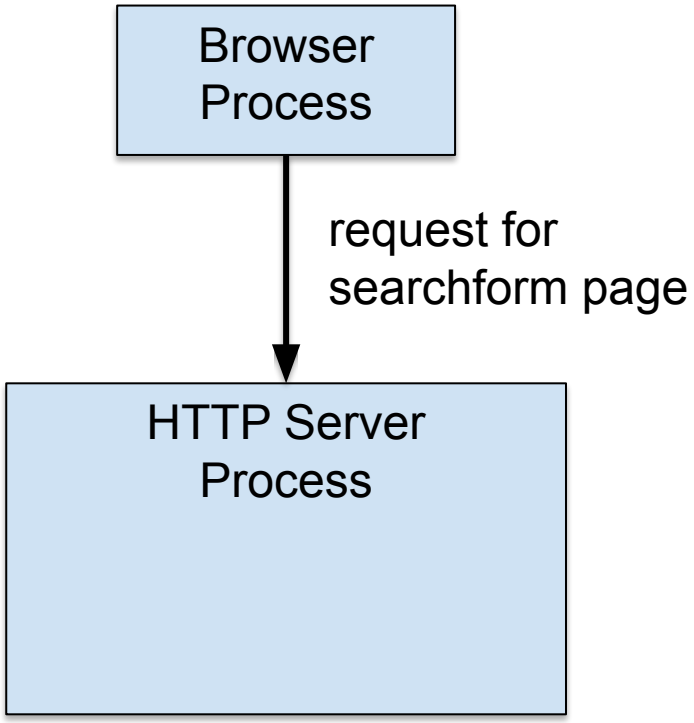
Option 1 (cont.)

Web App Architectures

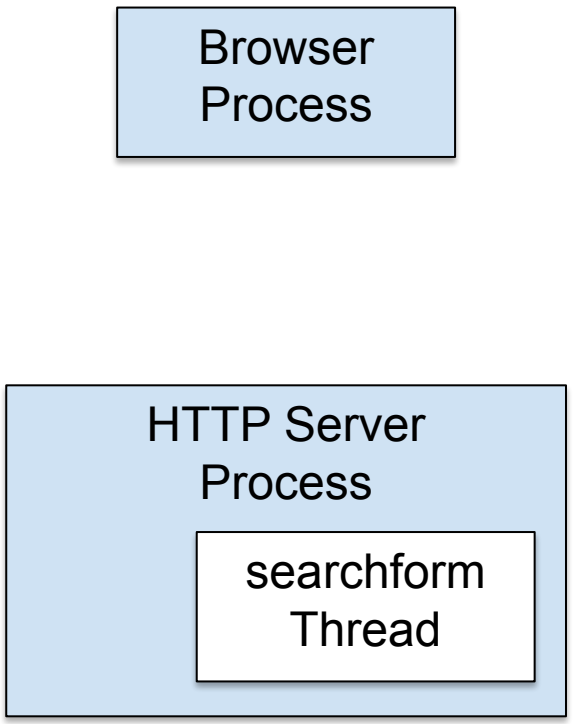
- Consider Option 2…
  - Web app **spawns a child thread** to handle each HTTP request

Web App Architectures

Option 2

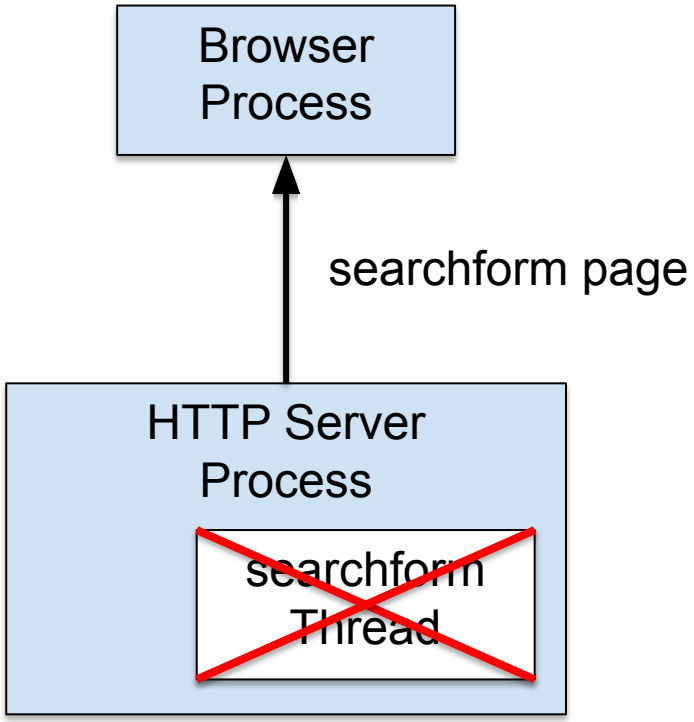


Web App Architectures



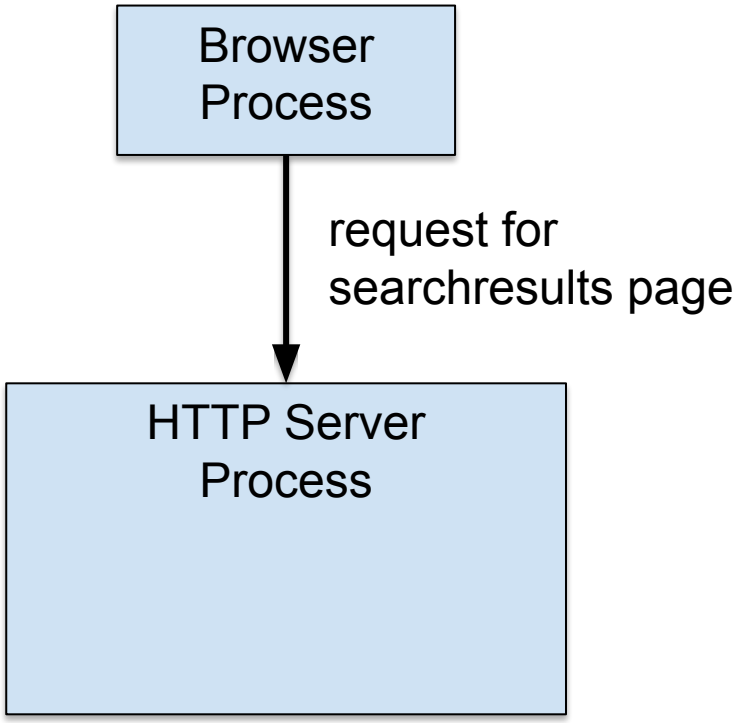
Option 2
(cont.)

Web App Architectures



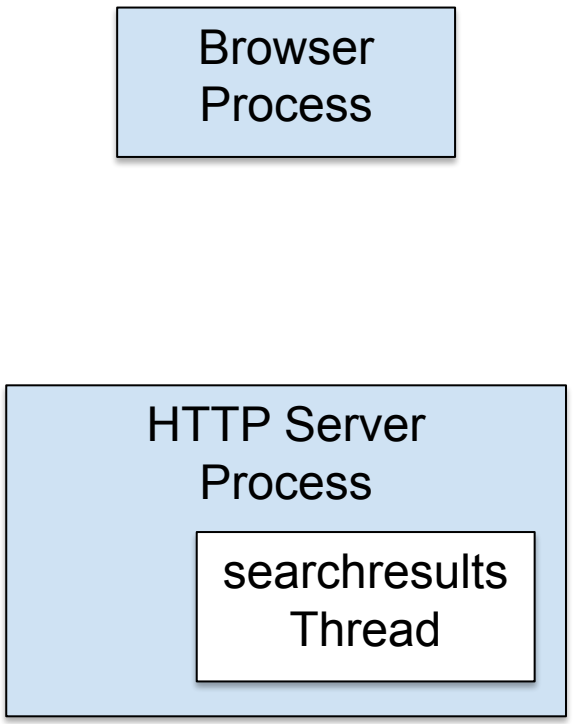
Option 2 (cont.)

Web App Architectures



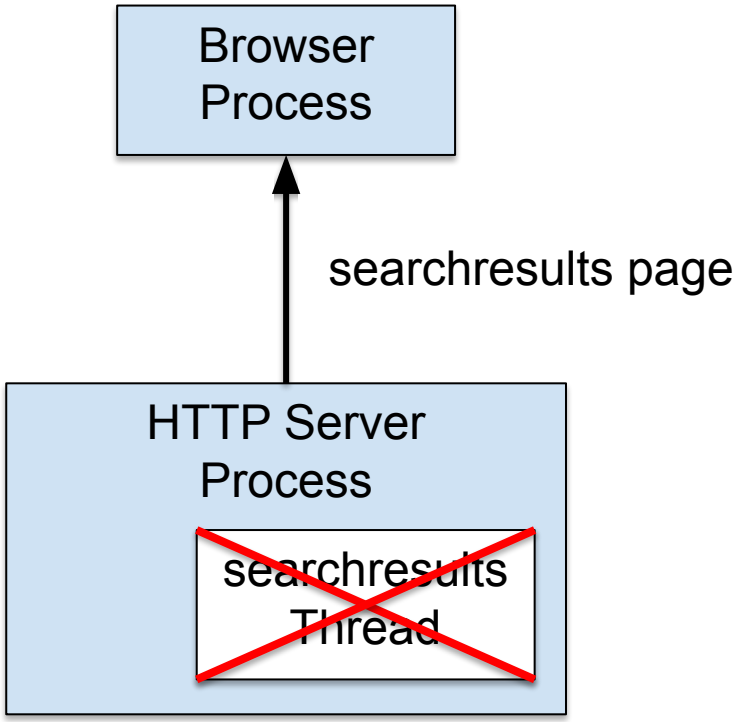
Option 2 (cont.)

Web App Architectures



Option 2
(cont.)

Web App Architectures



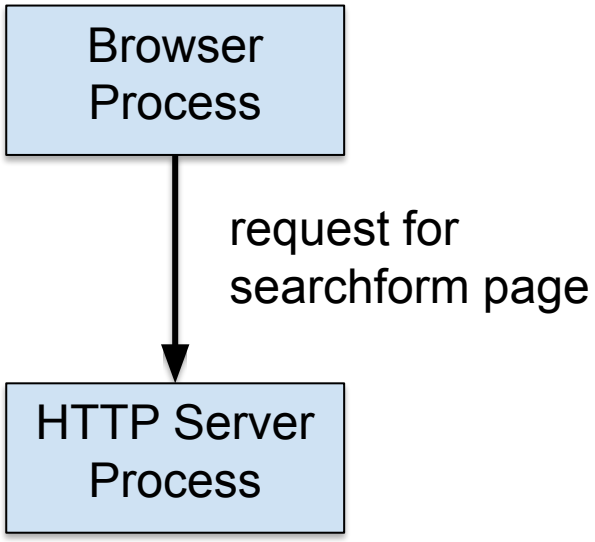
Option 2 (cont.)

Web App Architectures

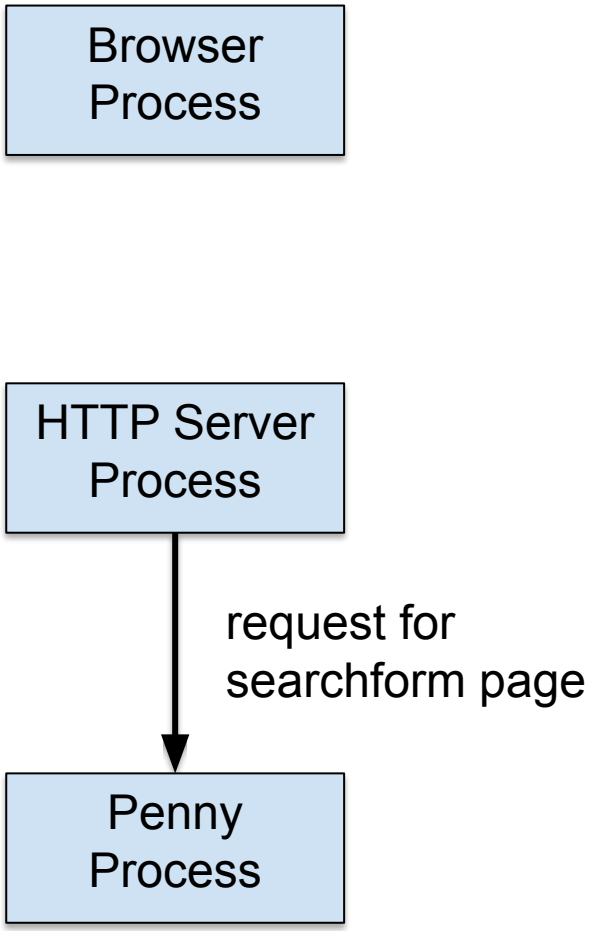
- Consider Option 3…
  - Like Option 1 (CGI)
    - HTTP server forks processes
  - Unlike Option 1 (CGI)
    - Server forks one process for each **application**, not for each **request**

Web App Architectures

Option 3

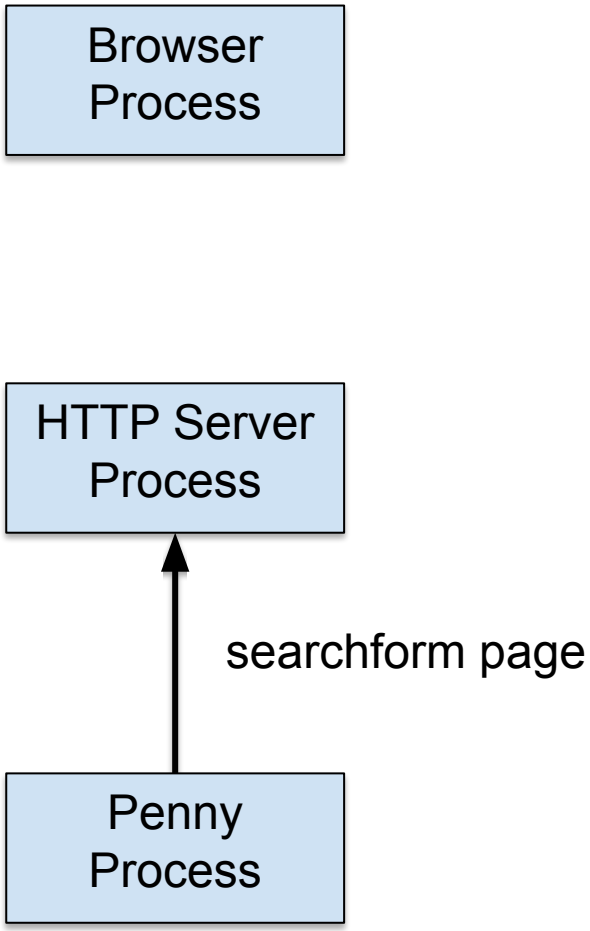


Web App Architectures



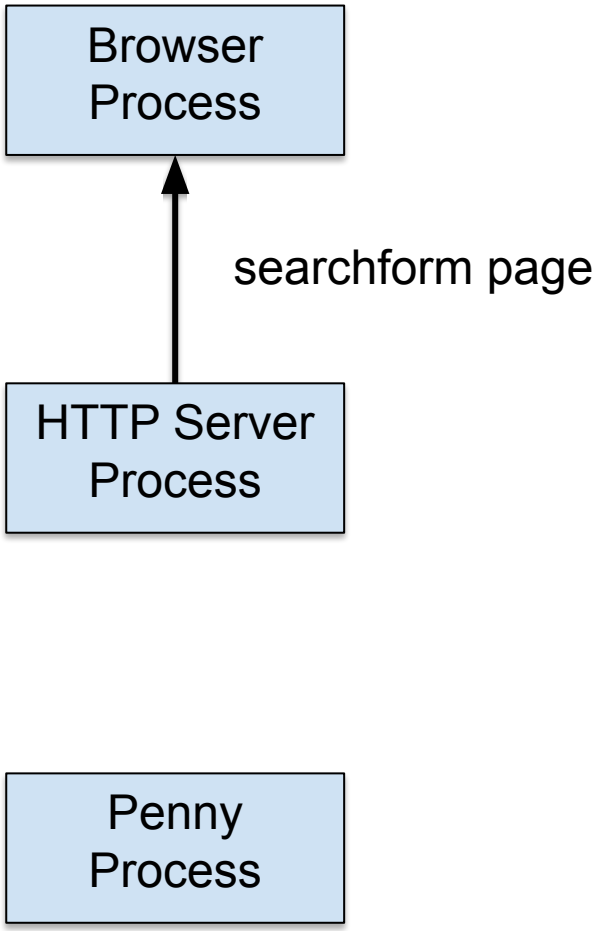
Option 3 (cont.)

Web App Architectures



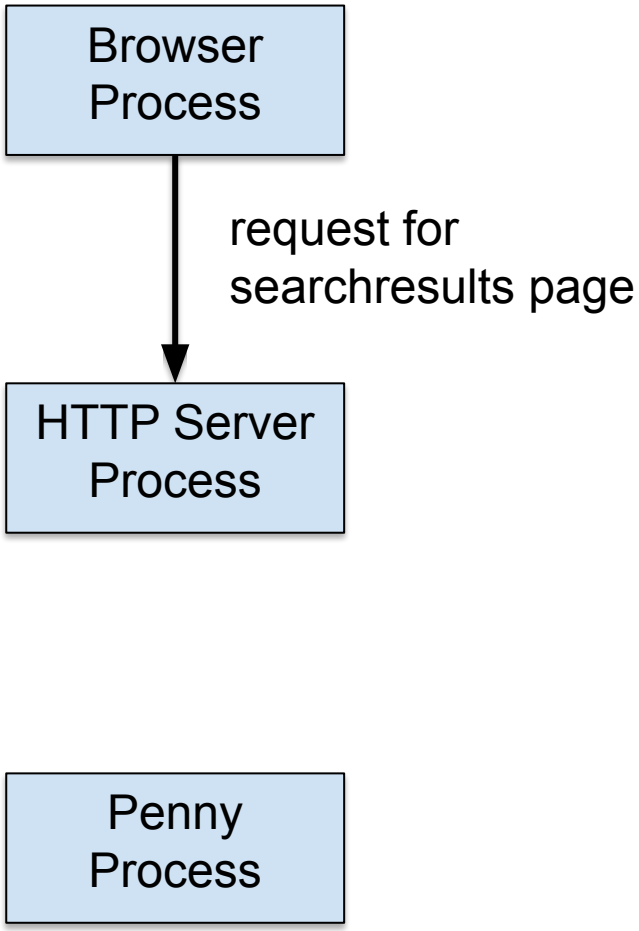
Option 3 (cont.)

Web App Architectures



Option 3 (cont.)

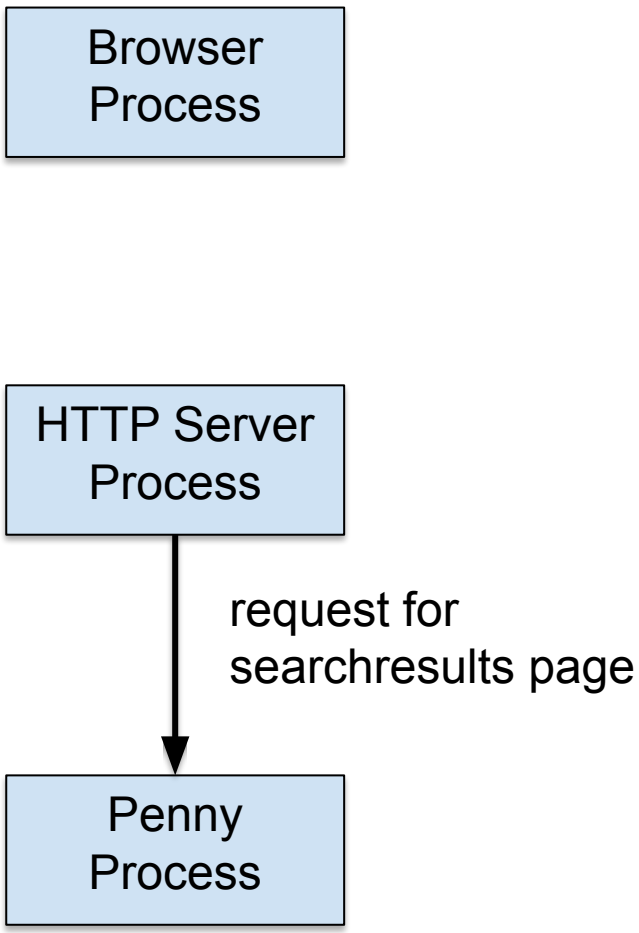
Web App Architectures



Option 3 (cont.)

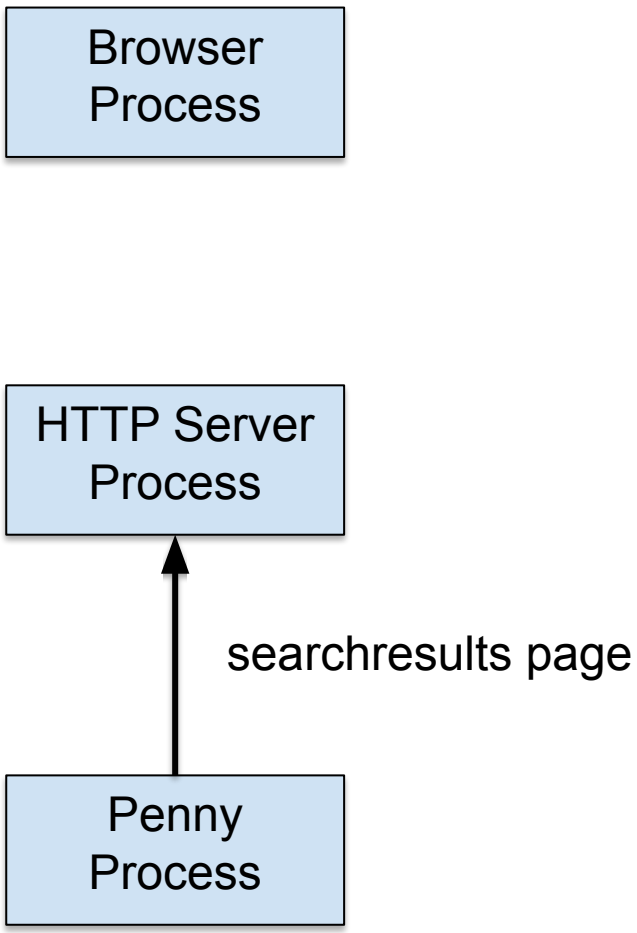
Web App Architectures

Option 3
(cont.)

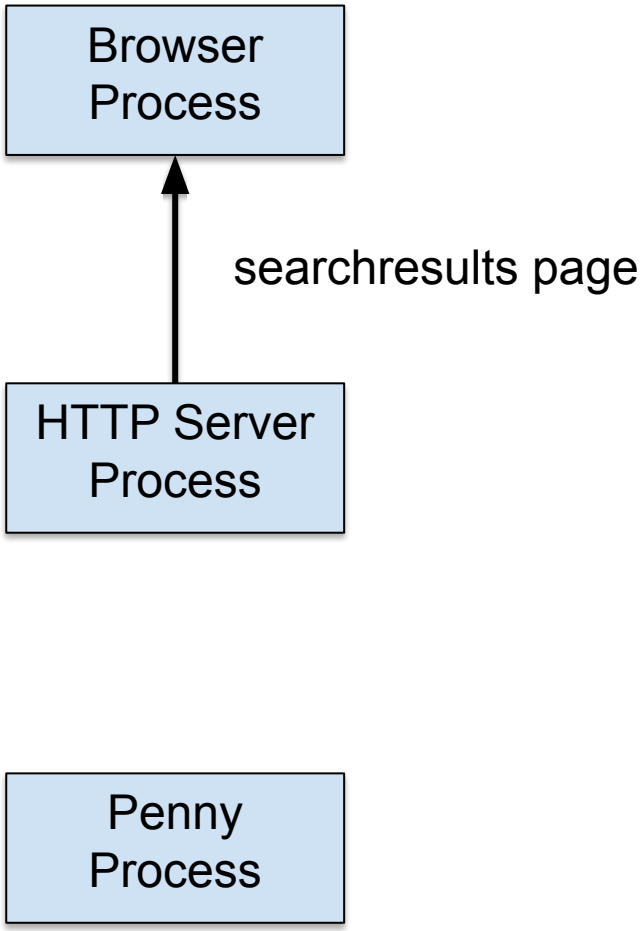


Web App Architectures

Option 3
(cont.)

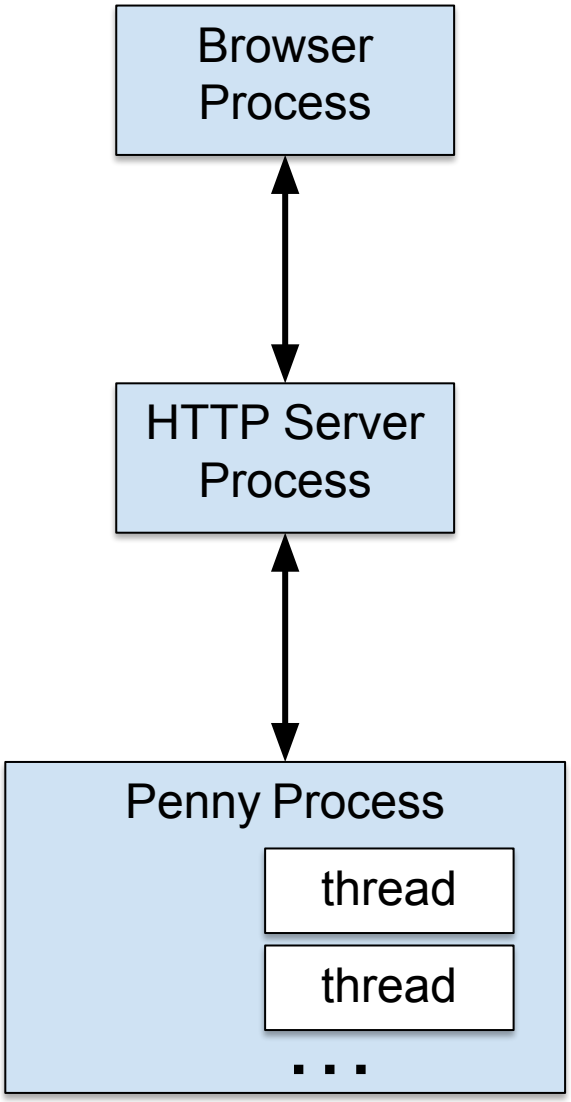


Web App Architectures



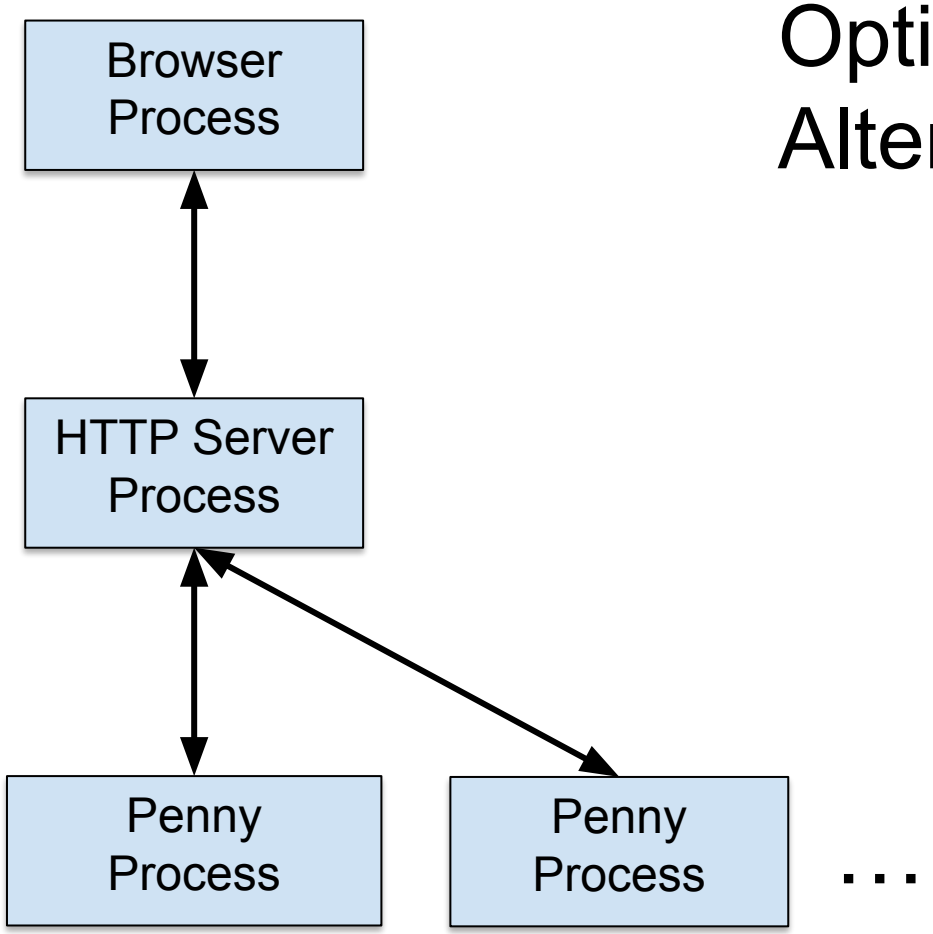
Option 3 (cont.)

Web App Architectures



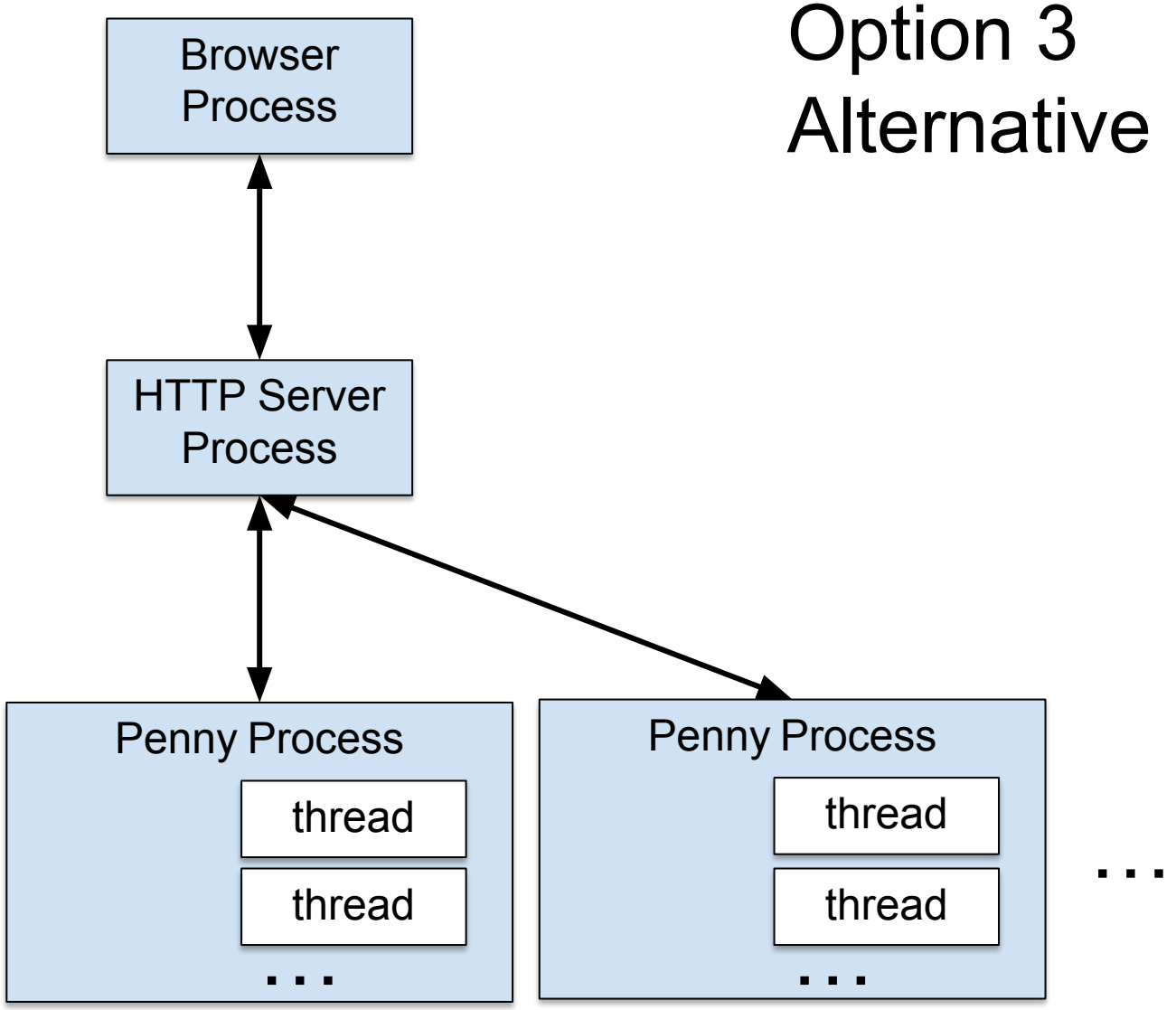
Option 3
Alternative

Web App Architectures



Option 3
Alternative

Web App Architectures



Option 3
Alternative

Web App Architectures

- Option 3 assessment:
  - (pro) Faster than Option 1 (CGI)
  - (pro) More flexible than Option 2
  - (pro) More secure than Option 2

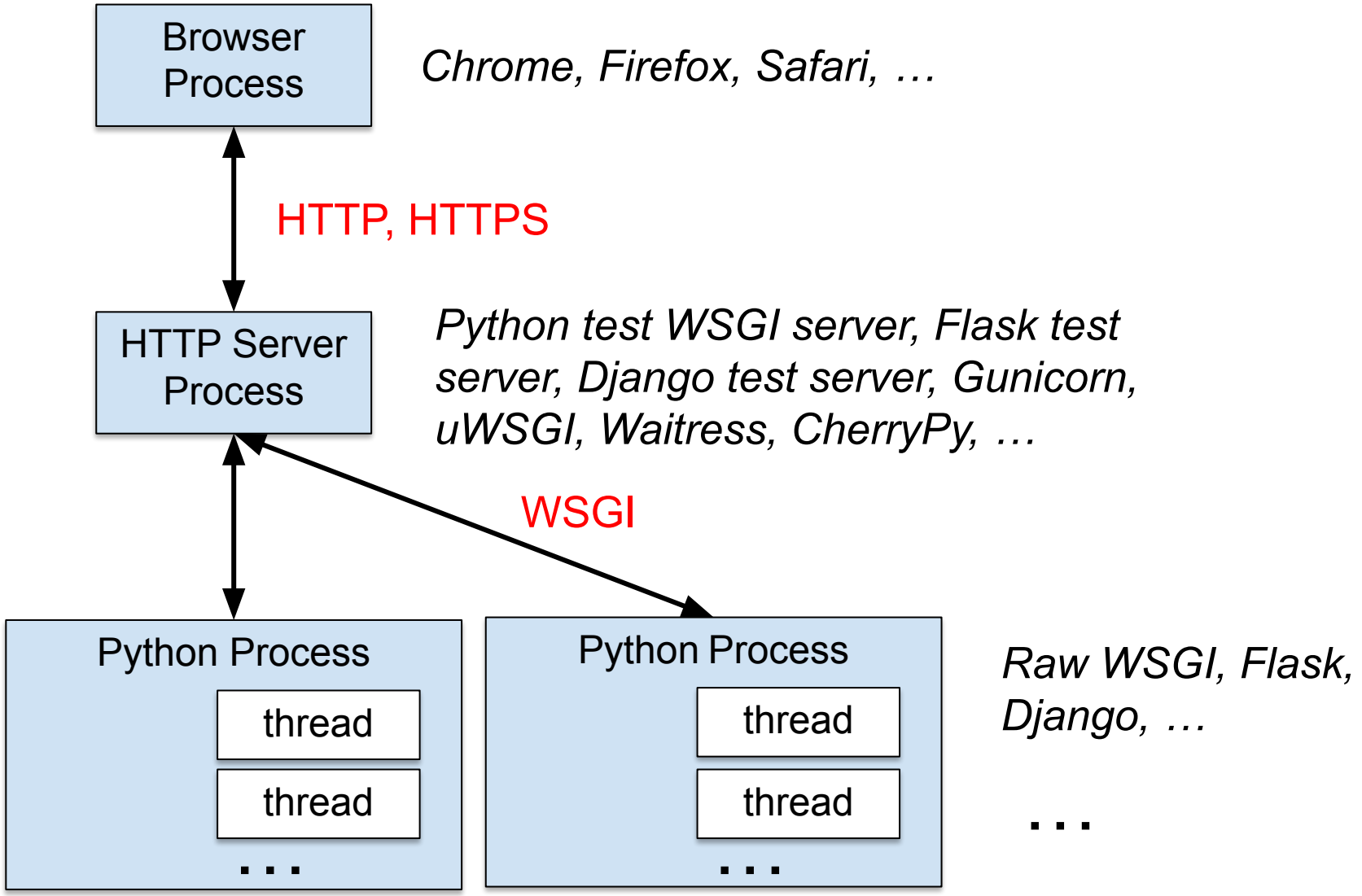
Agenda

- Web application architectures
- **Python WSGI programming**

Python WSGI Programming

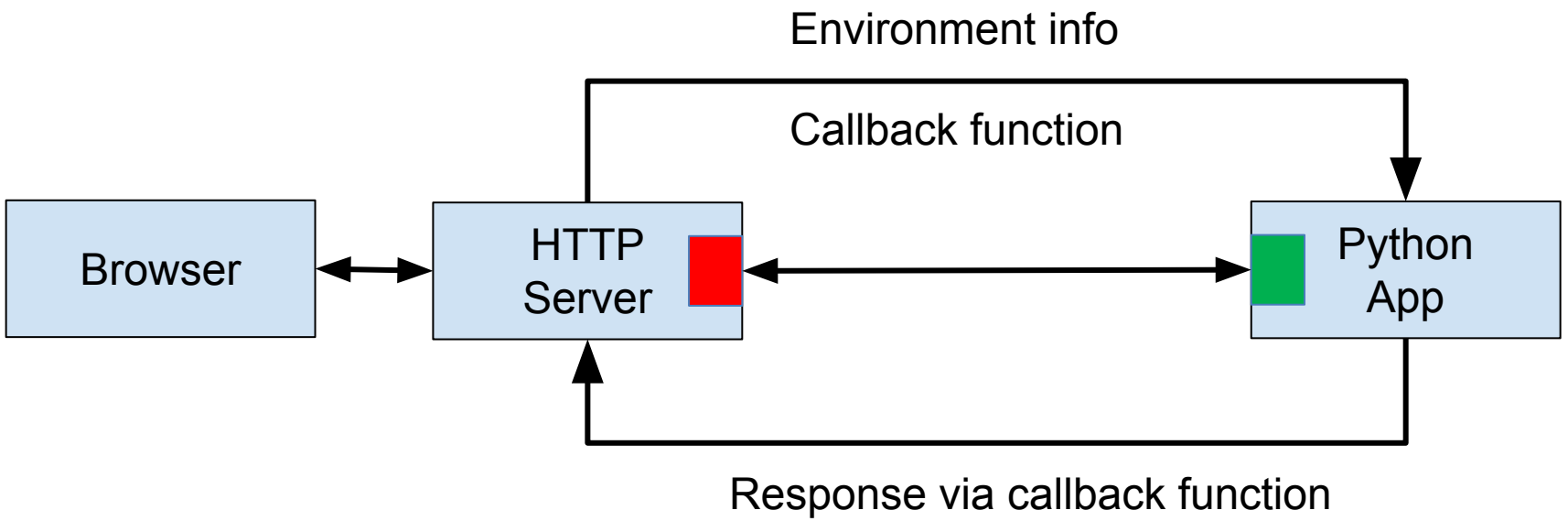
- ***Web Server Gateway Interface (WSGI)***
  - A standard interface for HTTP servers to communicate with Python applications

Python WSGI Programming



Python WSGI Programming

The Python WSGI Architecture

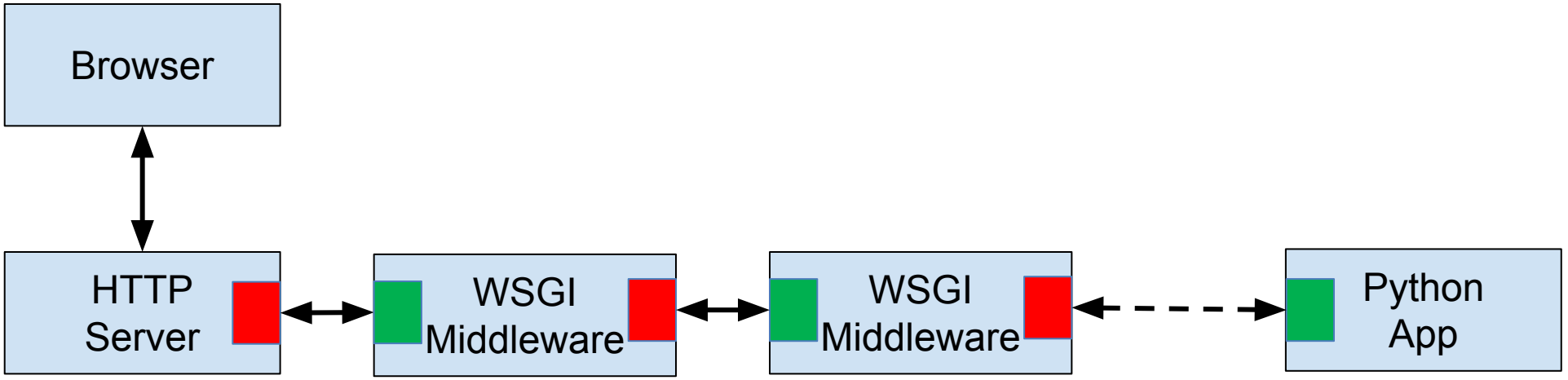


WSGI server-side
Defines messages Python app can send to HTTP server

WSGI client-side
Defines messages HTTP server can send to Python app

Aside: WSGI Middleware

WSGI allows ***middleware***



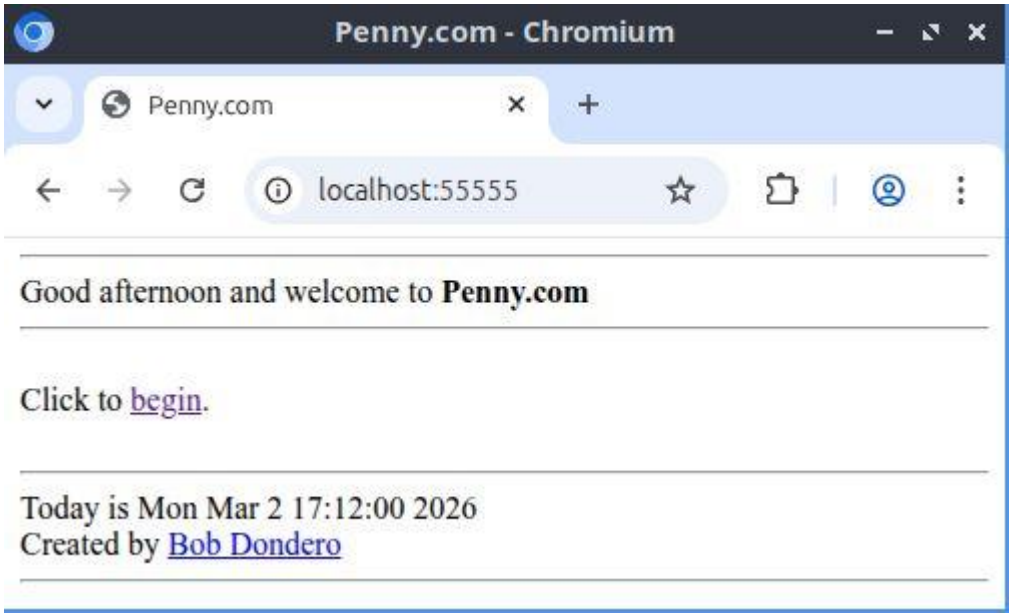
Python WSGI Programming

- See **PennyWsgi** app

```
$ python runserver.py 55555
Listening on port 55555
```

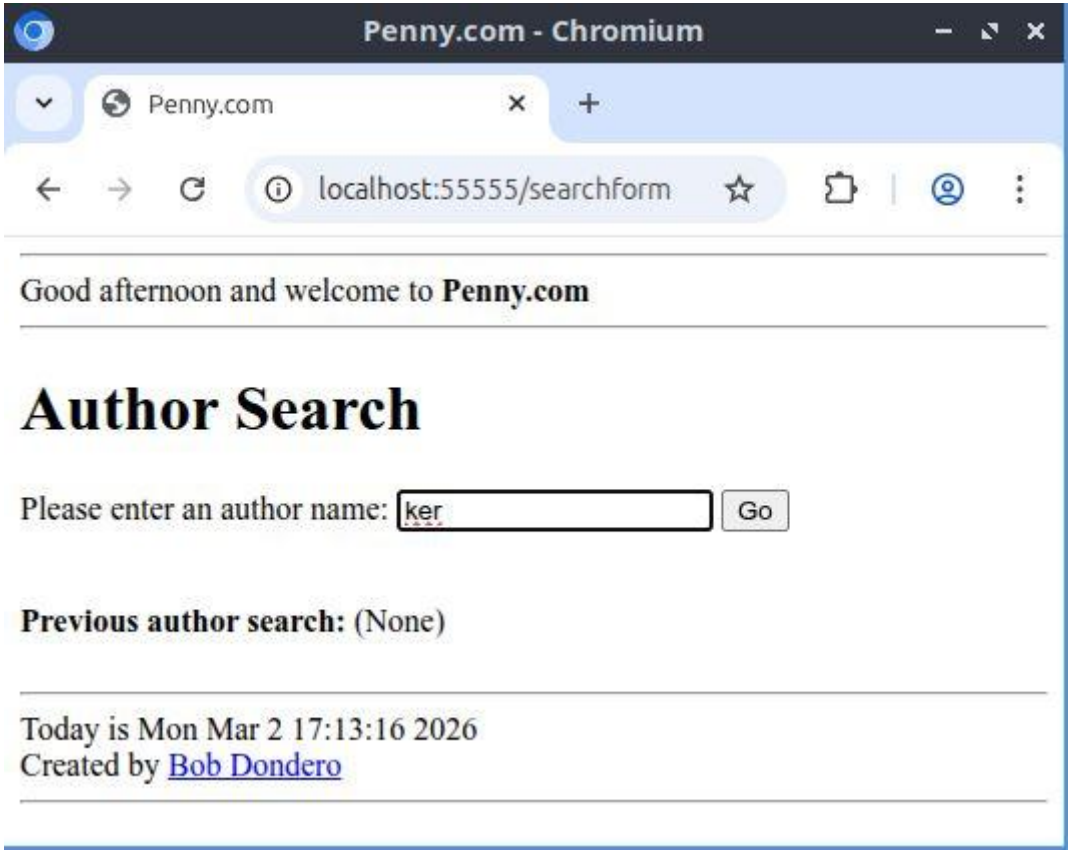
Python WSGI Programming

- See **PennyWsgi** app (cont.)



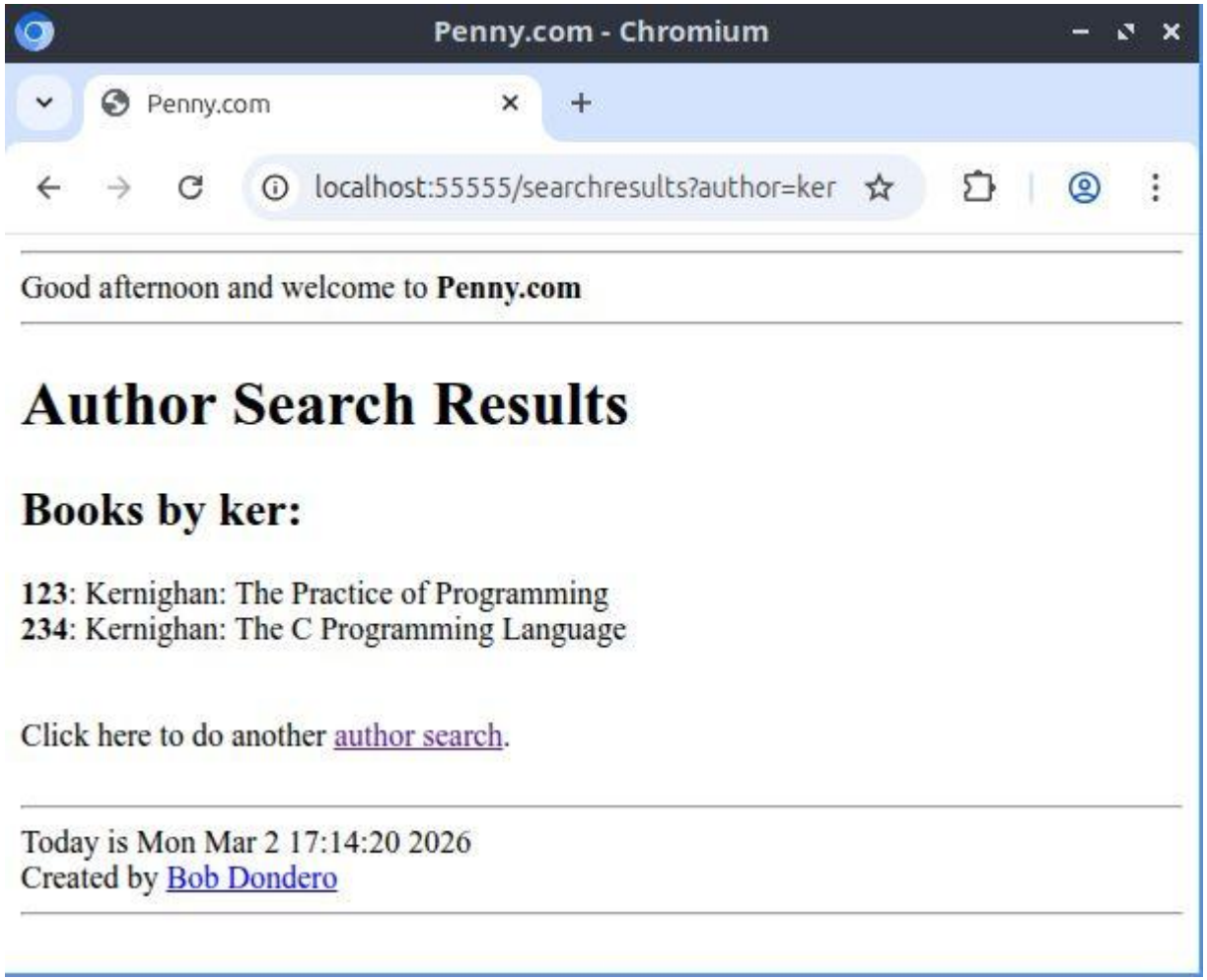
Python WSGI Programming

- See **PennyWsgi** app (cont.)



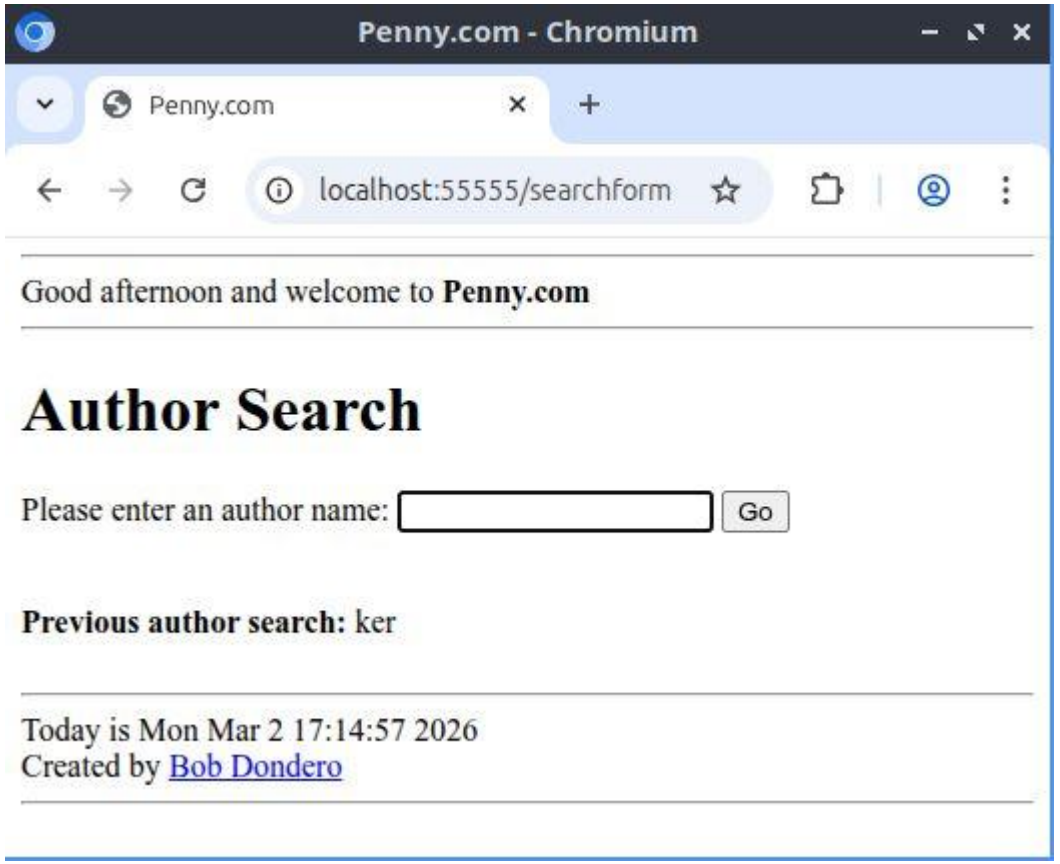
Python WSGI Programming

- See **PennyWsgi** app (cont.)



Python WSGI Programming

- See **PennyWsgi** app (cont.)



Python WSGI Programming

- See **PennyWsgi** app (cont.)
  - **runserver.py**
  - penny.sql
  - penny.sqlite
  - database.py
  - parseargs.py
  - **common.py**
  - **penny.py**

Python WSGI Programming

- WSGI vs. CGI:
  - (con) More complex
  - (pro) Faster
  - (pro) Separates URLs from **file** names
    - Allows use of “pretty” URLs
- WSGI also separates URLs from **function** names
    - Could improve maintainability

Lecture Summary

- In this lecture we covered:
  - Web application architectures
  - Python WSGI programming