Web Servers and Web Apps

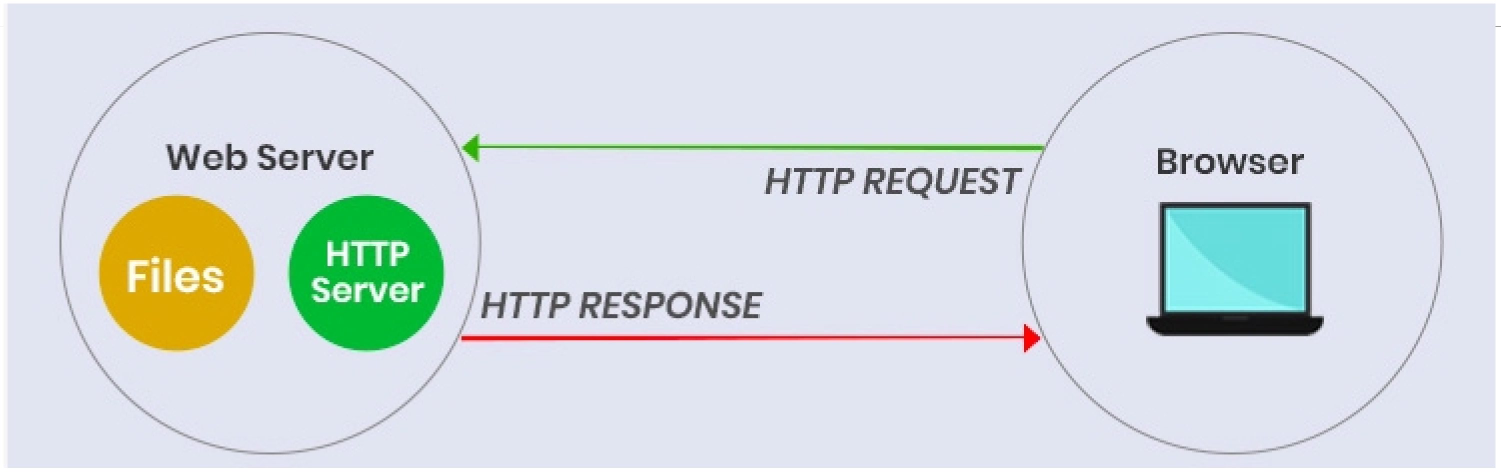
What a Web Server Does

- Receives web requests & sends replies
- Can serve **static content** (web page in a file, images, etc),
- Can create **dynamic content** using applications written in PHP, Python, Java, other; typically using a back-end processor for a specific language
- Handles security certificates and secure sessions
- May perform authentication, but many web apps and web frameworks do this themselves.

Static vs Dynamic Content

https://yoursite/index.html	static file
https://yoursite/media/logo.png	static file
https://yoursite/media/xmas.mp3	static file
https://yoursite/media/intro.webm	static file
https://yoursite/index.php	dynamic page from PHP
https://amazon.com/books/	dynamic page of best selling books
https://yoursite/polls/	list of polls generated by Django (Python)

Web Server for Static Content



Return content from files, based on the request URL.

Very little processing needed.

May perform authentication & authorization.

May process aliases, perform redirects, URL rewriting.

May perform cookie handling & session management.

Most Popular Servers: Apache Httpd & Nginx

Web Server for Application

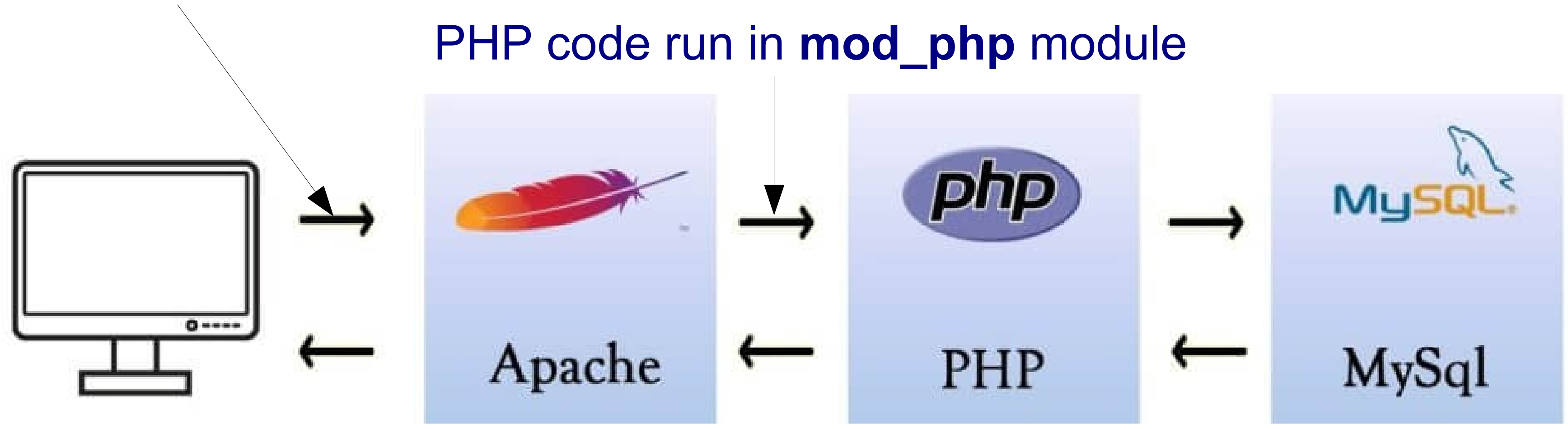
To create dynamic content, a web application uses "scripts" or applications in a programming language.

The web server forwards the HTTP request to the application running in a separate thread or process.

The app returns a response (or an i/o *stream*) that the server returns to the web client.

Web Client: **`GET /store/products.php`**

PHP code run in **mod_php** module

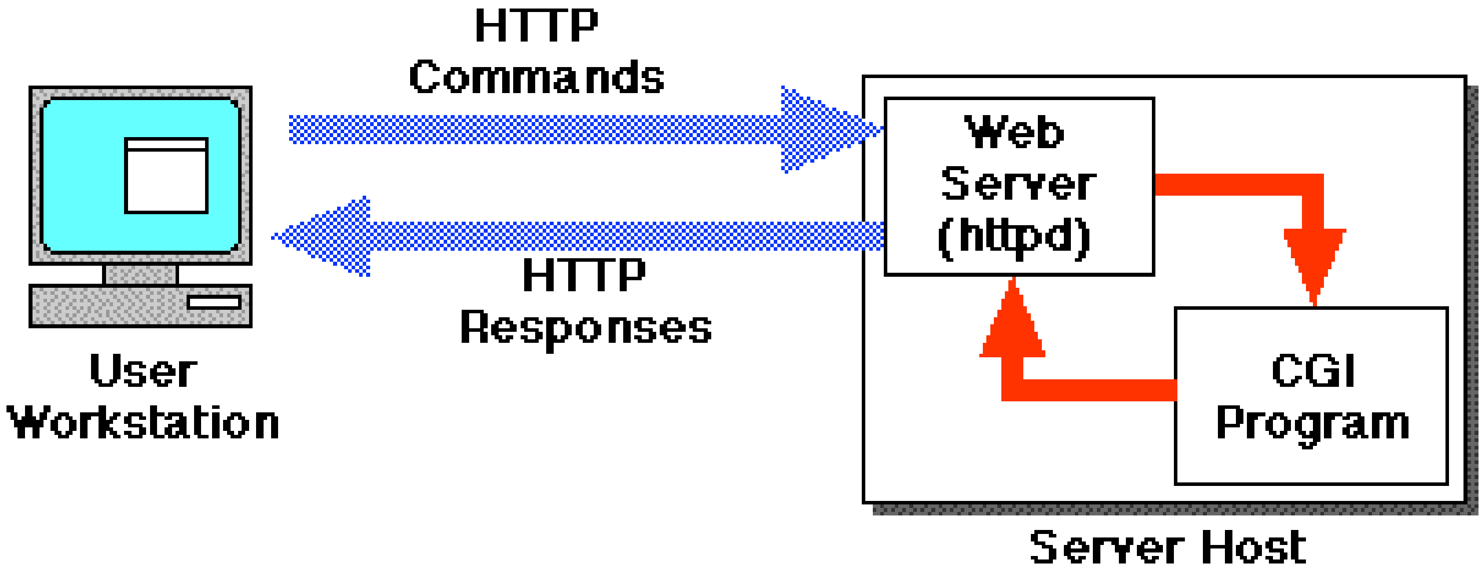


Common Gateway Interface

The original standard for invoking scripts or apps from a web server was CGI.

Each CGI request was run as a separate process, and it was the CGI app's responsibility to manage sessions, parse requests, etc.   Very low performance.

CGI scripts (apps) often written in Perl, C, or Bash script.

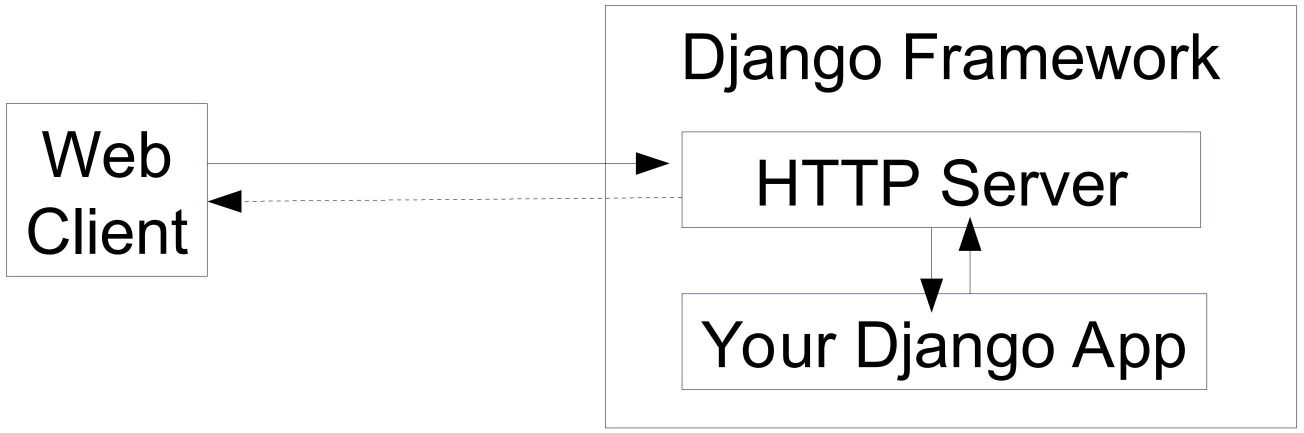


Django App in Development Mode

Django includes a built-in HTTP server, written in Python.

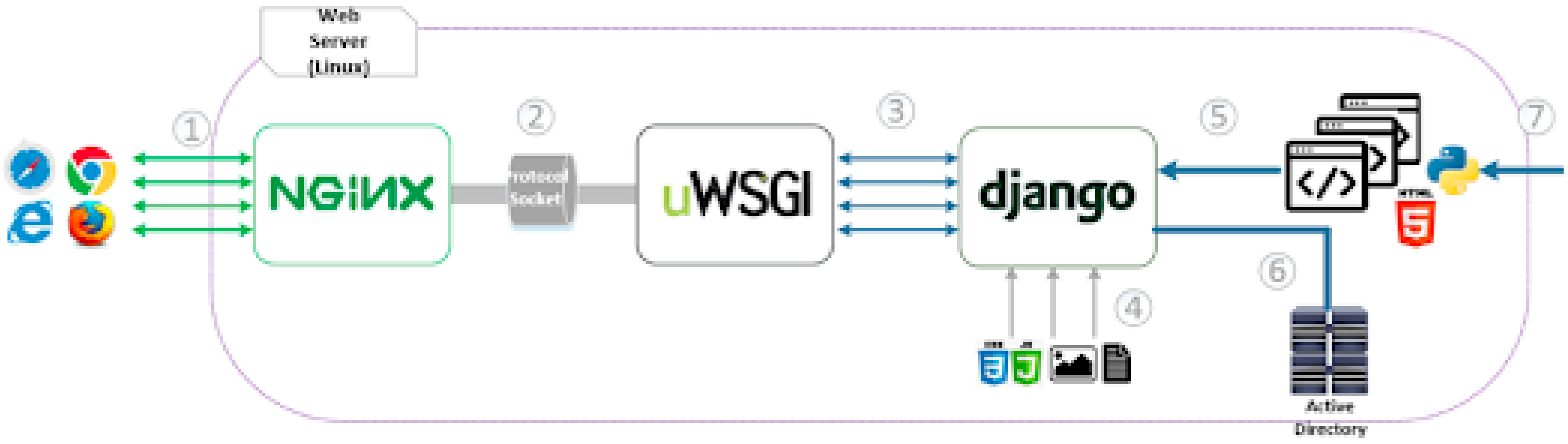
The server parses requests and passes them to the Django application.

This is *low performance* and not for production use.



*Running in **Python** (of course)*

Django App in Production



(1) Web server receives http requests. It handles requests for static content itself.

(2) Web server forwards http request to a WSGI container. WSGI is a standard interface to Python web apps, uWSGI is a web app container that supports WSGI.

(3) Django receives the request via wsgi, processes it, and returns an HTTP response

Web App Containers

A Web App Container executes or "serves" web applications and provides services to the app.

Python web app containers use the WSGI interface:

Gunicorn

uWSGI

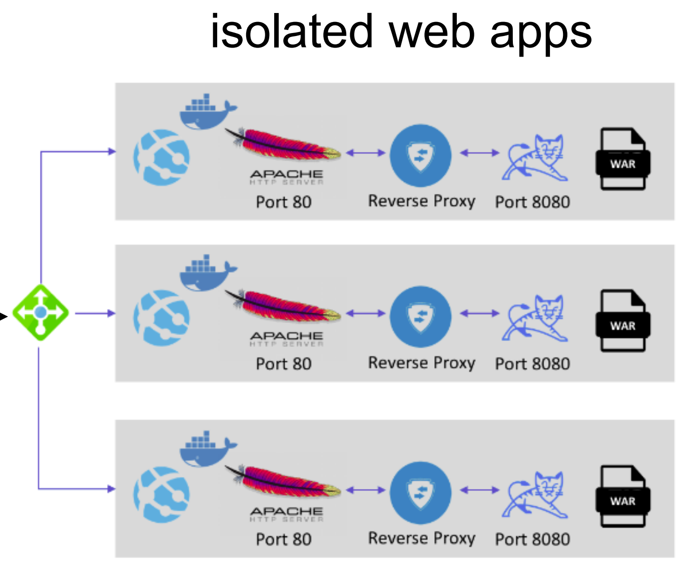
Java is a popular language for web apps because there are so many excellent Web App Servers & Containers for the Java web app standard (JEE), and the spec defines useful services for web apps.

Containers and VMs

For reliability and scalability, many web apps are run in Docker "containers" or on virtual machines (VMs).

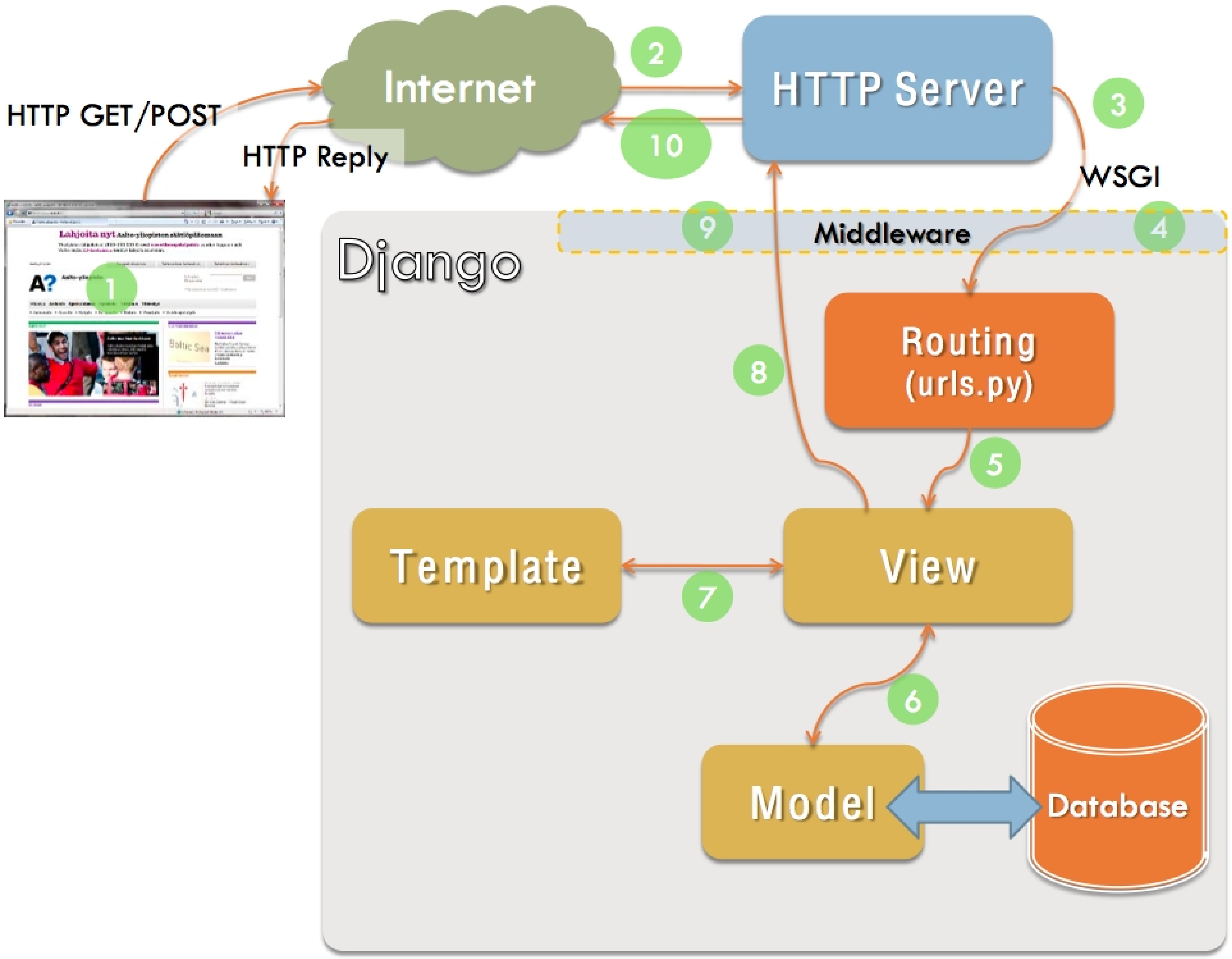
- Web app in a Docker container.
  - The app and its dependencies are bundled in an "image" run in a Dockor container.
  - New instances can be started quickly to handle high loads.

Http request



- Deploy in a Virtual Machine.
  - Even stronger isolation.

Django Request Processing



Source: https://devopedia.org/django

The Most Popular Web Servers?

nginx

Apache httpd

OpenResty

Cloudflare

Monthly surveys:

- *Netcraft Server Survey https://news.netcraft.com*
- https://www.datanyze.com/market-share/web-and-application-servers/

What Software Does SCB Use?

1. Go to SCB EasyNet at: www.scbeasy.com
2. Choose English or Thai.
3. Look at the URL on the logon page.

https://www.scbeasy.com/v1.4/site/presignon/index.asp

What web framework does SCB use?

Experts say that you should <u>not</u> expose implementation in URLs. A better URL would be:

https://www.scbeasy.com/login

Extra Material

You can skip the following slides.

Content Delivery Networks and Caching are services to improve performance on the world wide web.

Content Delivery Networks

- Akamai, Digital Island, etc.
- A network of servers that replicates content (such as images and video) at many different sites around the world.
- When a web browser requests content (image, video), the CDN delivers it from the closest site!
- It does this by cleverly directing your web browser to a CDN host that is closest to your location.

CDN Example

You visit www.cnn.com & the web page contains images. Each image has a URL like this:

```
<img src="https://cdn.cnn.com/assets/images/sports.gif">
```

- "cdn.cnn.com" refers to a CDN provider like Akamai.
- Your web browser sends a DNS request to get the IP address of "cdn.cnn.com".
  (It does this the first time only, then remembers the IP address for a while.)
- "cdn.cnn.com" has many IP addresses -- their DNS server returns the IP address of the server **closest to your location**.
- Your browser gets the images from the CDN server closest to you. All the CDN servers have identical copies of all the content.

Web Caching

- Caching is <u>critical</u> to performance of the web
- Multiple levels of caching:
  - client (web browser caches content)
  - server (manually configured cache)
  - gateway (uses a transparent Cache Engine)
  - network (CDN, cooperating caches)

Cache Engines

- Harvest (free)
- Squid (free)
- Cisco Cache Engine (based on Harvest)

Why Web Caching?

- Decrease use of network bandwidth
- Faster response time
- Decrease server load
- Security and web access controls (auth, blocking)