Faculty of Engineering and Applied Science

SOFE 4790: Distributed Systems

CRN 43524

Homework: Web Server Software

Alexander Campbell

100703650

Software Requirements of a Web Server

- Internet Protocols: A family of communication protocols that are used by the internet for programs and machines to communicate to each other
- Sockets: A socket is an end-to-end communication device that uses a port assigned to it on the host computer and an IP address.
- HyperText Transfer Protocol: This is one protocol that uses text to transfer data between machines or programs. It involves the client sending a request for some information and the server responding to it with the information or some other request depending on the success of retrieving the information.

At what layers of the OSI models does this web server system operate? What is the role of HTTP, DNS, HTML in the described system?

In a web server, technically all layers of the OSI model are utilized in some form. At the lowest level, the Physical layer, there are actual physical devices that transfer the information when any request is being made. At the Data Link layer we have the formatting of the data occurring, where the request is put into a packet that can be sent off to the server it's requesting from by the Physical layer. This is also where connections to the server from the client may be terminated. In the Network, Transport, and Session layers these information packets that are being sent between the client and the server can have these packets resent and ensure there's an open communication channel between the client and server. Finally in the Presentation and Application layer we have the information that has been sent to the client from the server being displayed in some manner.

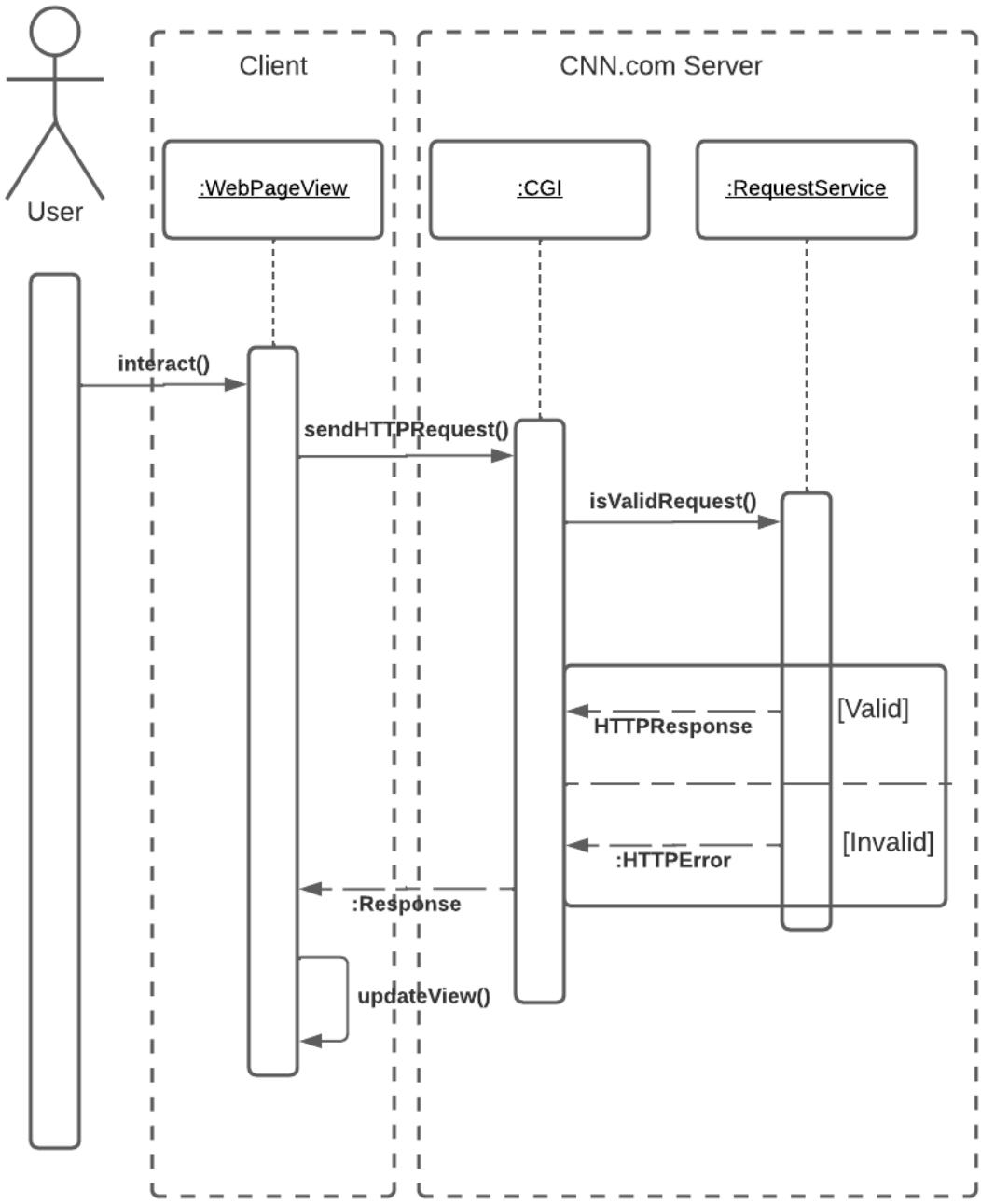
Both HTTP and DNS are both in the Application layer. HTTP at its core are just information requests from a client to server, these requests allow the user to interact with the server and create requests thus belonging to the Application layer. DNS is a domain name service, which maps names, such as google.com, to connectable IP Addresses. This is also done at the Application layer since the client is requesting the specified domain name that is converted to an IP address when communicating with the server.

HTML is at the Presentation layer of the OSI model. The reason is HTML is the formatted data request that the client can view in a readable format.

How are transparencies provisioned within the described image sharing system?

Replication transparency is very important in a web server. It ensures that the image sharing system's data is readily available but also hidden from the users that want to access the data that corresponds to the web server. Access transparency is also provisioned where user's don't need to know the physical location of the web server, just the DNS.

Take CNN.Com for example, their web server updates the information displayed on each client? How can a web server do this? Draw UML sequence diagrams? What functions/ components need to be added to the described system?



What is 'synchronous messaging' or request reply messaging? What is the role of a web server in this type of messaging? How is that different from Asynchronous Messaging?

Synchronous messaging is a type of request-reply messaging in which a response is required from the recipient of a message. It can be used in many different scenarios, such as to send notifications or to provide updates on a process. The role of a web server in request reply messaging is to provide a service to users who are requesting data from it. The user sends out information to the server and then receives information back from it. Asynchronous messaging refers to a message-passing system in which messages are not sent or received in real time, meaning that messages may be sent at any time with no guarantee the recipient has received them. This is different from synchronous messaging in which the recipient is guaranteed to receive it as there is a response after receiving the information.

What is the role of a web server in a Queue-Based Load Leveling Pattern?

In Queue-Based Load Leveling Pattern, web servers are the components that have the responsibility to take the request load of the application, receiving requests and then queuing them up. This way, they can distribute the load to multiple servers which will avoid overloading any single server and provide better performance to users.

What can you do with CGI? What are the pros and cons?

A common gateway interface (CGI) is a software interface that allows programs on one machine to communicate with programs on another machine by receiving and forwarding data.

Pros:

- Allows for communication between networks, which can lead to new innovations in the field of networking and distributed systems.
- Allows for the integration of heterogeneous systems, leading to more efficient use of resources and better performance.
- Provides a standardized way to access data from different sources, which can be useful in many applications such as web browsing or accessing files on your computer.

Cons:

- Introduces some security risks because it provides an easy way for hackers to access your data if you have not taken steps to secure it properly.
- Requires changes in the network's architecture that may not be feasible for some networks or organizations.

How can we enhance CGI? Read about three-tier architecture and multi-tier architectures. How did this functionality evolve from CGI?

CGIs can be enhanced by utilizing several specialized gateways that have separate data forwarding capabilities. This later evolved into multi tier architectures where data management, processing, and presentation are separated in a program. CGIs are needed in this process to receive and forward information to various machines to the various layers.

A web server is just one piece of the "enterprise architecture", what are other pieces?
The components that make up enterprise architecture are as follows:

- Data management
- Networking
- Databases and data storage
- Operating systems
- Virtualization

How can such architecture handle very large amounts of data? Where would you fit a database? How can the operating system support such a web server?

The enterprise architecture can handle large amounts of data through the use of data management components. These components allow for efficient and safe storage of data such that the business can easily scale their databases up or down depending on their need.