



Dr Markus Hagenbuchner

markus@uow.edu.au

CSCI319

**- Distributed Systems -
Distributed Web-based Systems**

Distributed Web-Based Systems

Lecture notes based on the textbook by Tannenbaum

Study objectives:

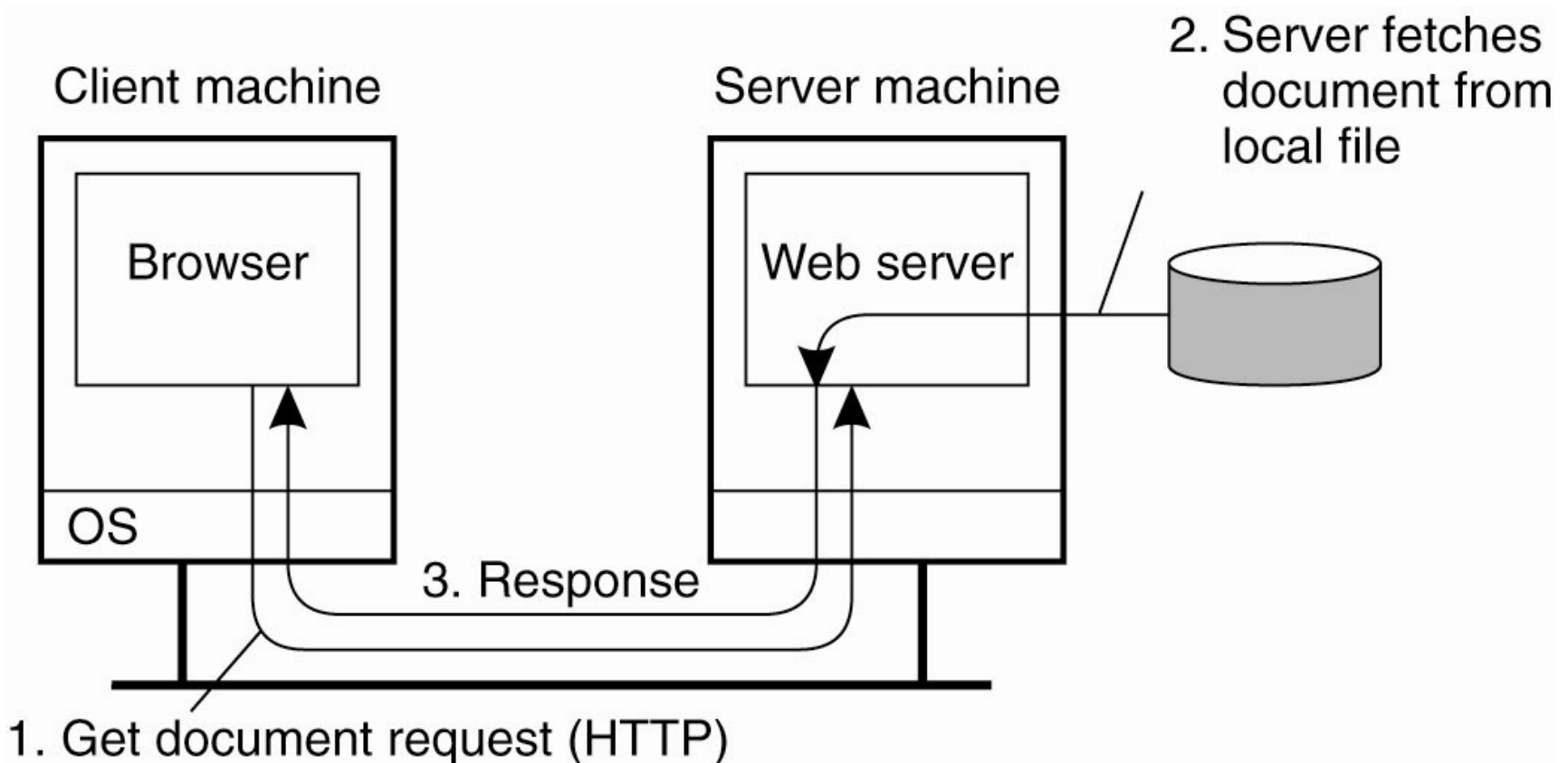
1. Understand the role of distributed web based systems
2. Explain how the eight design principles can be adopted in the realization of distributed web based systems.
3. Understand how scalability is achieved in apache and in content aware web server clusters.
4. Explain persistent connections and content aware caching.
5. Understand the HTTP Protocol

Content

- Basic Web based systems
 - Web documents and dynamic content
- Distributed Web servers
 - How to distribute services?
 - Architectures (i.e. server cluster)
 - Client and server processes
 - Also: Communication, Naming, Fault tolerance, Synchronization, Consistency and replication, Security.

Traditional Web-Based Systems

The overall organization of a traditional Web services:



Web Documents

A Web server specifies the type of a returned document by MIME type. Some common MIME types and subtypes:

Type	Subtype	Description
Text	Plain	Unformatted text
	HTML	Text including HTML markup commands
	XML	Text including XML markup commands
Image	GIF	Still image in GIF format
	JPEG	Still image in JPEG format
Audio	Basic	Audio, 8-bit PCM sampled at 8000 Hz
	Tone	A specific audible tone
Video	MPEG	Movie in MPEG format
	Pointer	Representation of a pointer device for presentations
Application	Octet-stream	An uninterpreted byte sequence
	Postscript	A printable document in Postscript
	PDF	A printable document in PDF
Multipart	Mixed	Independent parts in the specified order
	Parallel	Parts must be viewed simultaneously

Interactive slide

Mime:

How can the various document types be interpreted correctly by a browser?

What document types can be found on the Web?

Interactive slide

Mime: Multi-purpose Internet Mail Exchange

How can the various document types be interpreted correctly by a browser?

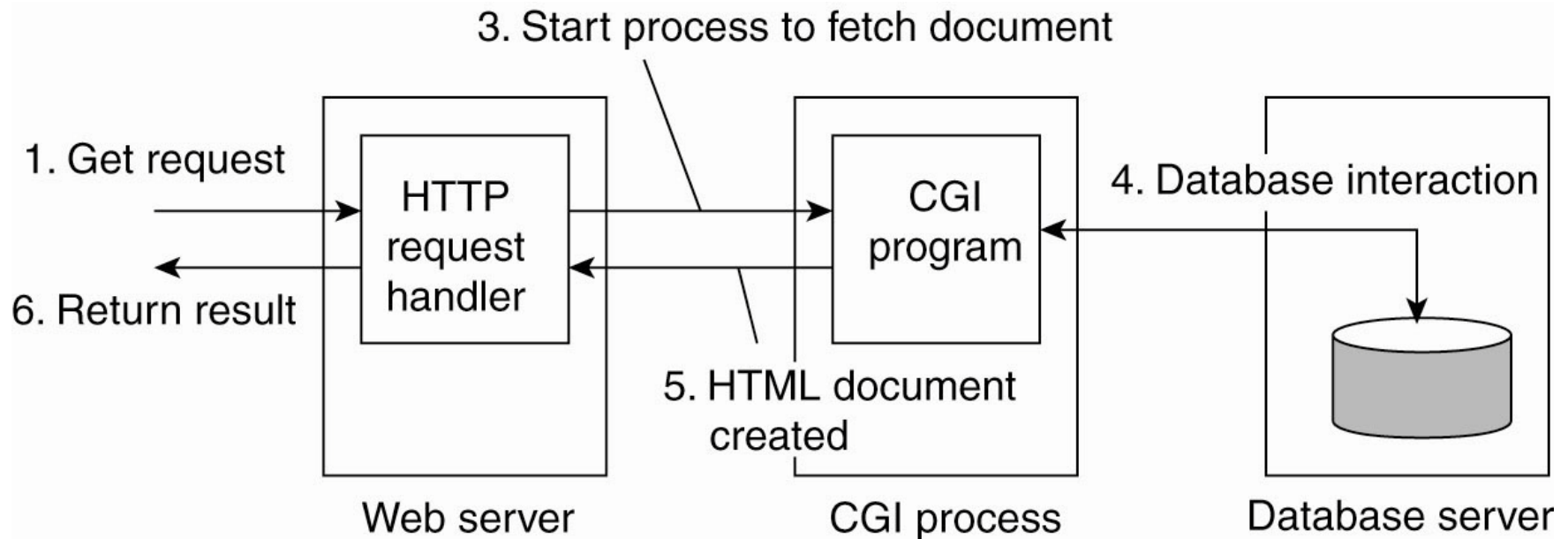
Through code migration (i.e. plugins loaded from a central code repository)

Types of documents on the Web?

Any type of digital information.

Multitiered Architectures

Modern Web services support dynamically created content. The principle of using server-side CGI programs:



Interactive slide

Meaning of acronym CGI?

What is CGI used for, and why is it useful?

Can a client request the CGI source code?

Interactive slide

Meaning of acronym CGI: Common Gateway
Interface

What is CGI used for, and why is it useful?

Serves as a transparent interface between client and data.

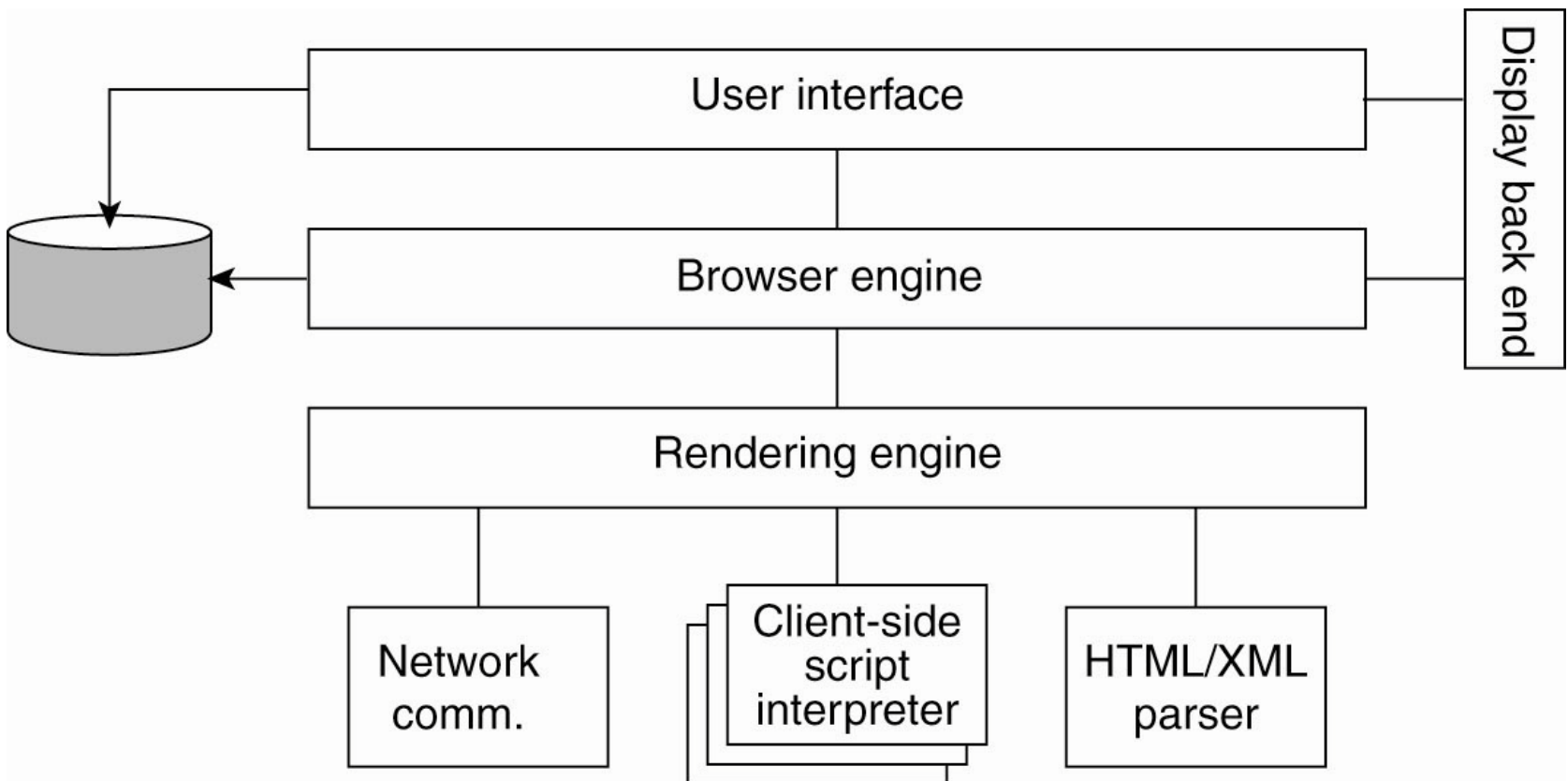
- Create personalized content dynamically from a template.
- Provide access control mechanisms.
- Convert document format.
- ...in a transparent fashion.

Can a client request the CGI source code?

No

Processes – Clients (1)

The logical components of a client side processes (i.e. a Web browser):



Processes – Clients (2)

Originally, Web browsers were designed to support HTTP only. A Web proxy process was introduced to allow the browser to access FTP content:



Nowadays, Web browsers support a large range of document formats due to the availability of code migration methods. But proxies remain popular due to additional functions that they can perform: authentication, caching.

Interactive slide

Purpose of proxy?

—

—

What else can the proxy be used for?

—

Explain how a proxy can address client-side security?

—

Interactive slide

Purpose of proxy? Allows to achieve

- transparency.
- Translation of global protocol to a local protocol.

What else can the proxy be used for?

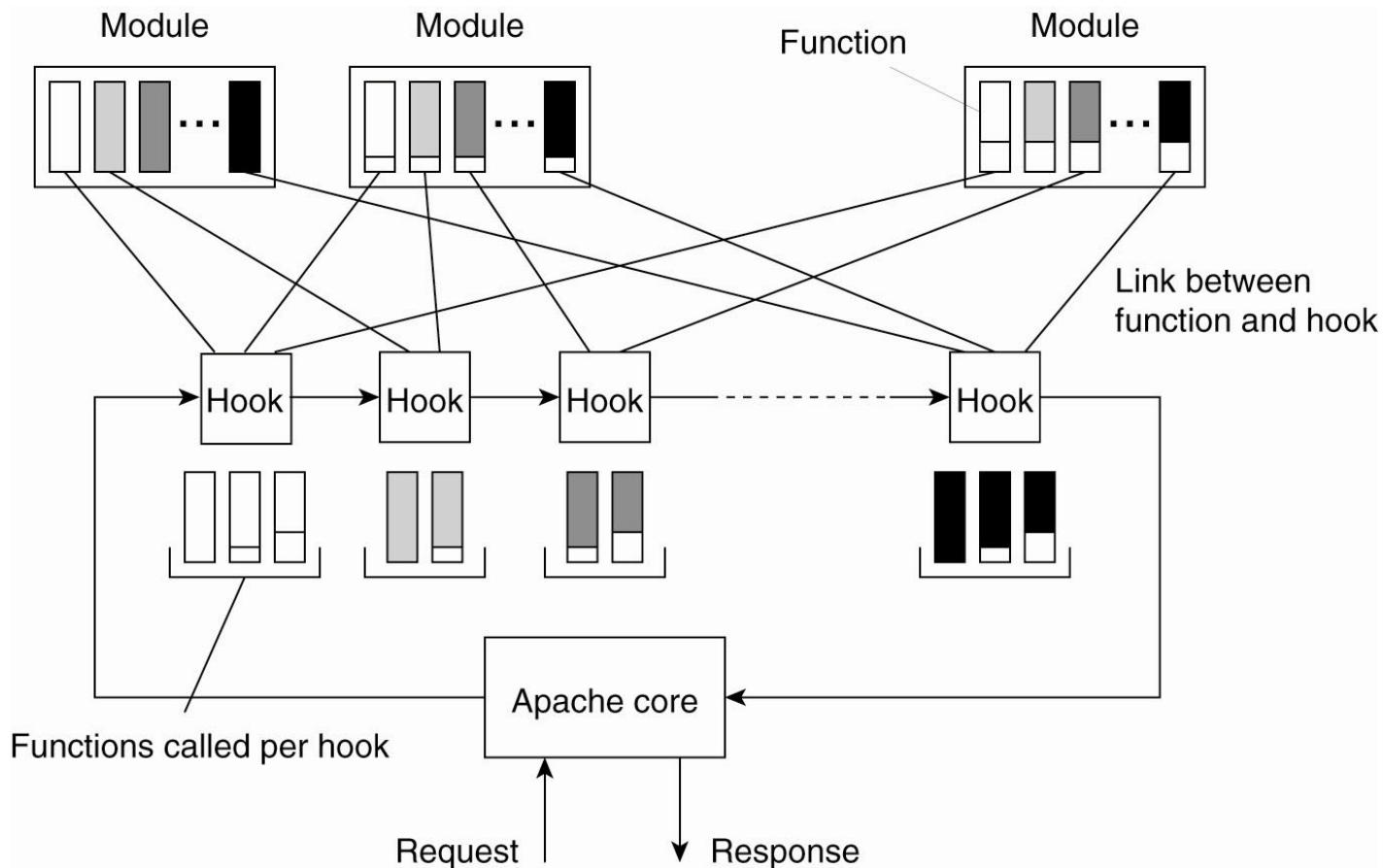
- Speed improvement through caching amongst several users.

Explain how a proxy can address client-side security?

- Can introduce an additional layer of security (authentication, firewall)

Server processes: The Apache Web Server

The general organization of the Apache Web server.



Interactive slide

Why apache?

Purpose of hocks?

What is achieved through the use of hocks?

Examples of what one hock may perform?

Interactive slide

Why apache? Scalable, extremely versatile Web server architecture.

Purpose of hocks? Each hock is responsible for executing a specific group of related functions that may need to be executed as a result of a request.

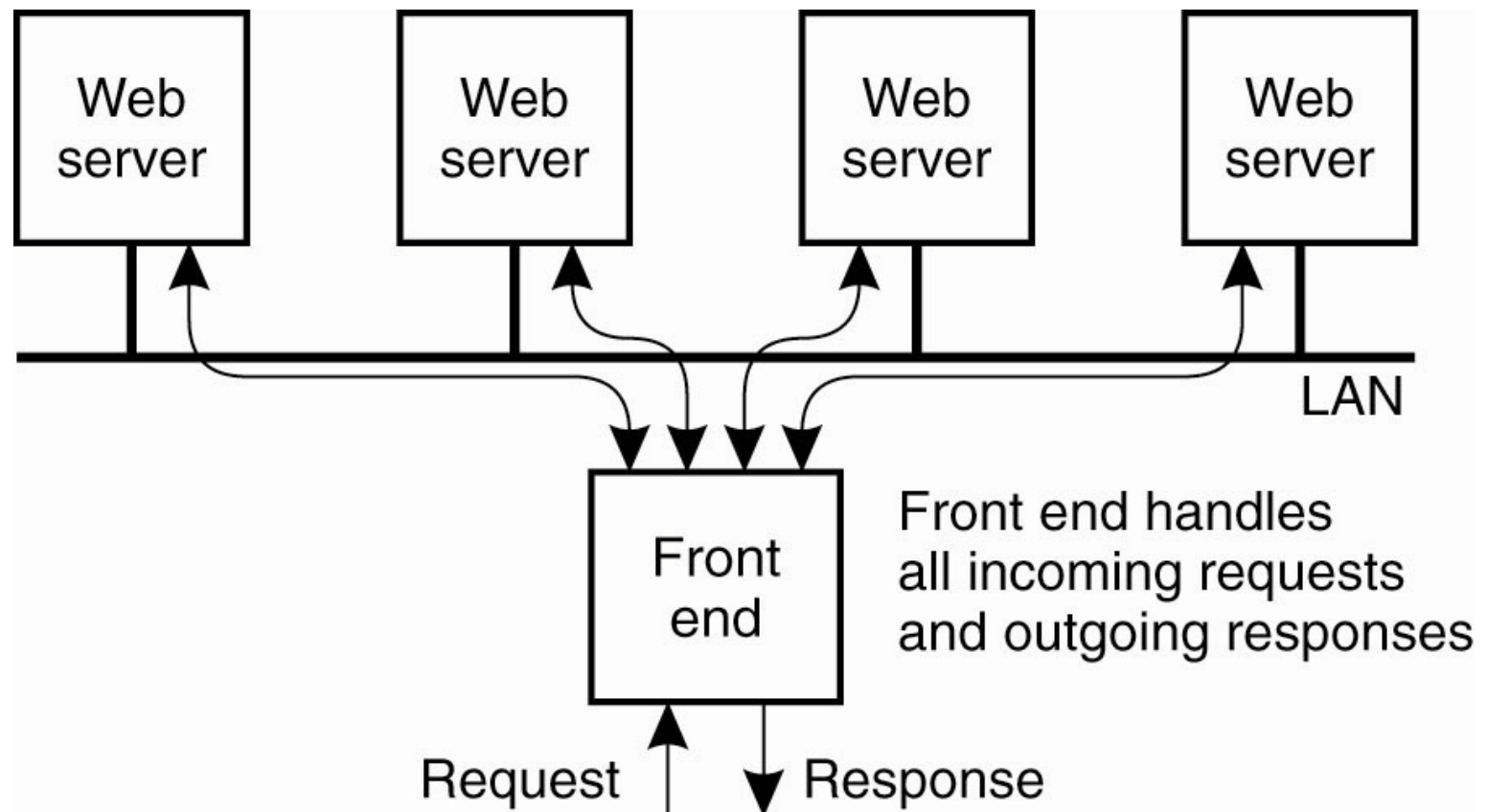
What is achieved through the use of hocks? Scalability through parallelism and pipelining of tasks.

Examples of what one hock may perform?

Resolve URL from a local file, write to a local log file, perform authentication, check access rights, read from a file, execute a CGI, etc.

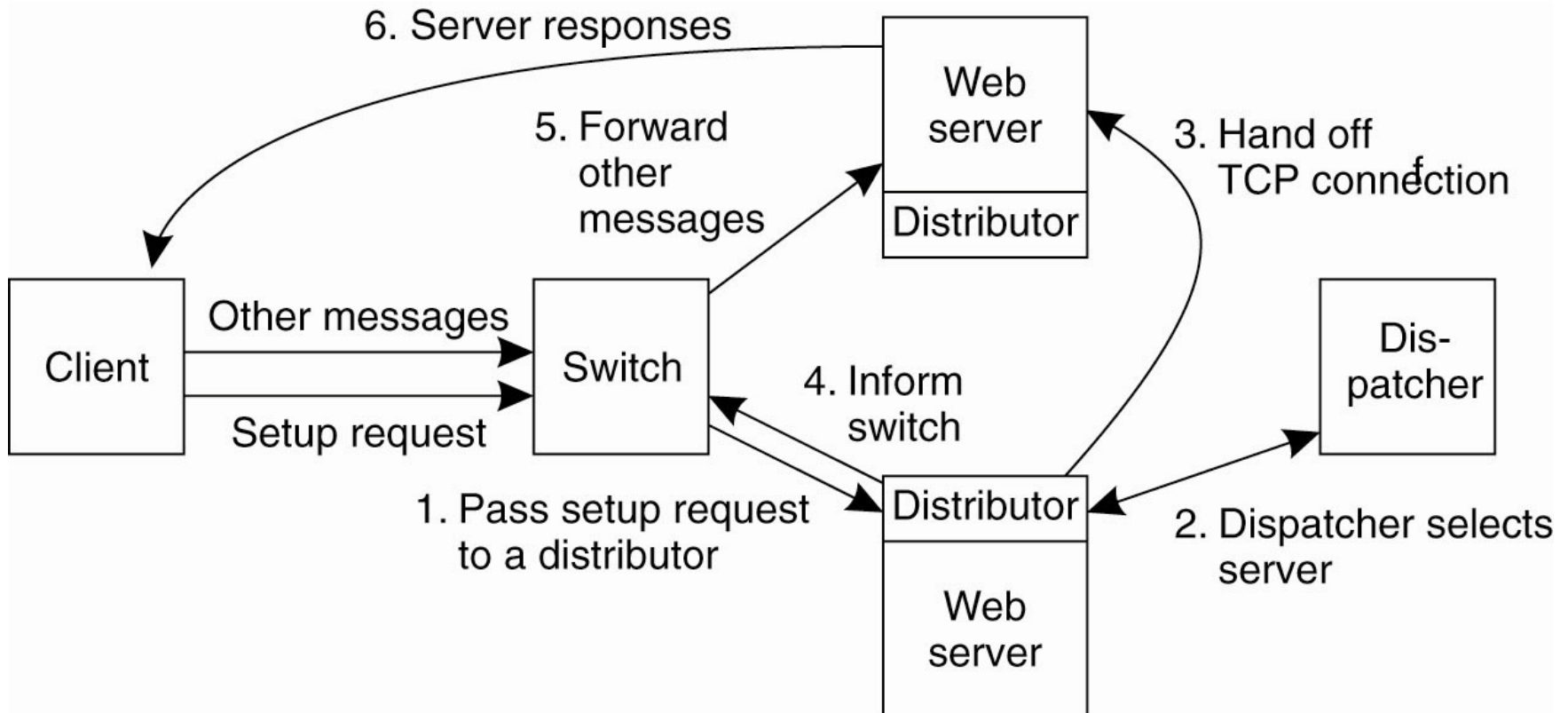
Web Server Clusters (1)

The principle of using a server cluster in combination with a front end to implement a Web service.



Web Server Clusters (2)

In a content-aware cluster of Web servers, the switch “knows” what a server has done before. Requests are relayed so as to maximise cache hits (to servers that have processed the same request before).



Interactive slide

Purpose of switch?

Purpose of dispatcher?

Purpose of distributor?

Scalability of the approach?

Interactive slide

Purpose of switch? Relay requests. Can aim at selecting a server which, for instance, handled a similar request in the past (to maximize chance of accessing cached data), or which is located closest to client.

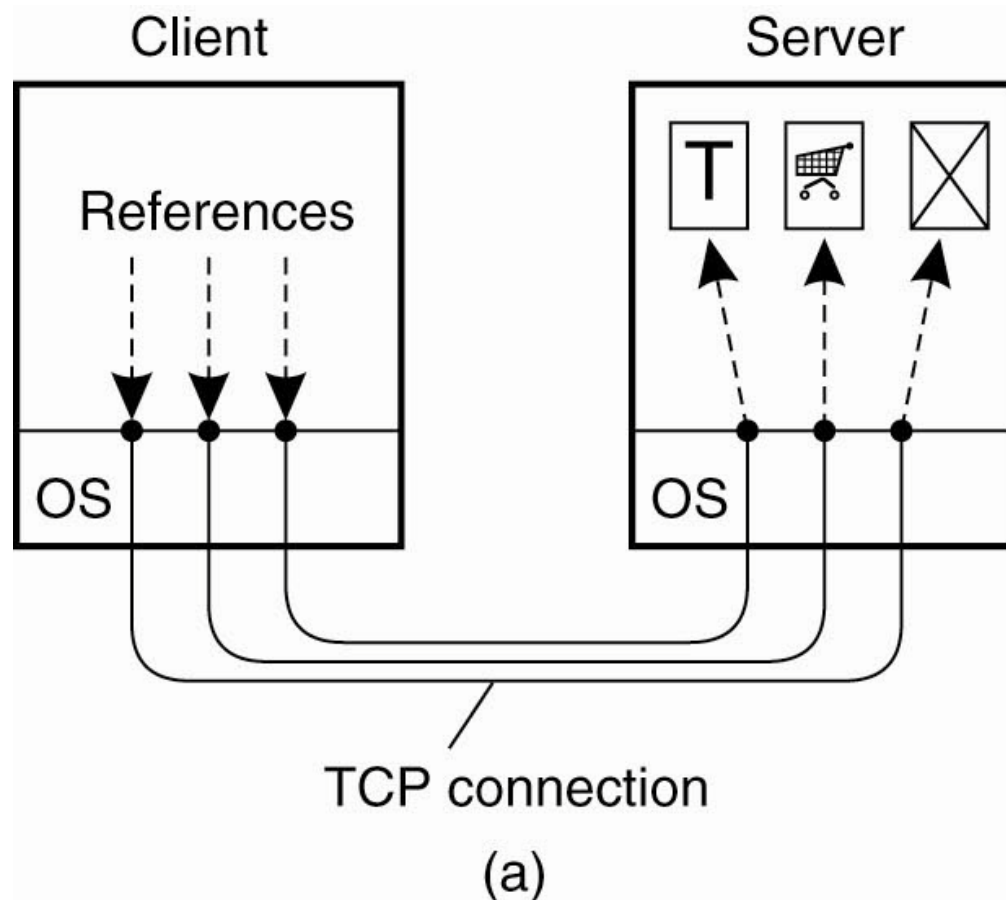
Purpose of dispatcher? Monitors availability of Web server, and selects an available one.

Purpose of distributor? First point of contact for the switch, reports to dispatcher.

Scalability of the approach? Performs automated load balancing on arbitrary number of nodes. Requests to servers with a high load are redirected to available servers.

Communication (1)

Example HTTP: When using non-persistent connections. A separate TCP connection is established to load every component of a Web document (i.e. load embedded images):



Interactive slide

What is a non-persistent connection, and where can this be found?

When are non-persistent connections becoming inefficient?

How could this be overcome?

Interactive slide

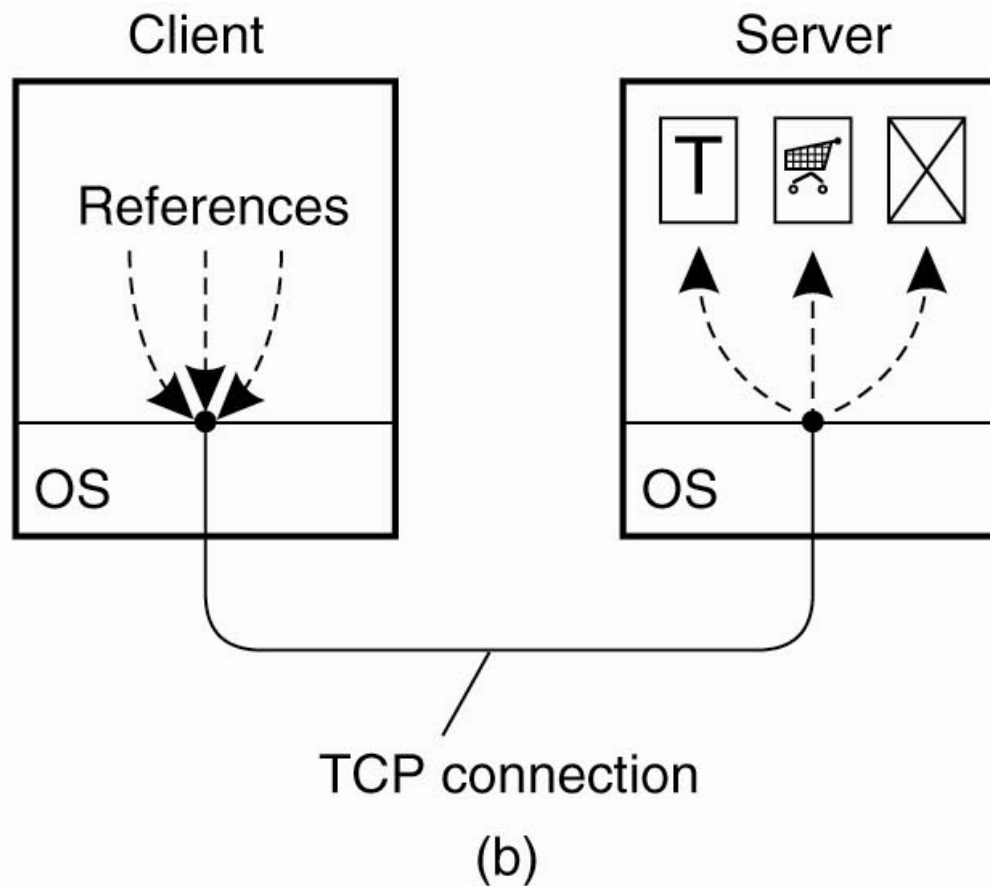
What is a non-persistent connection, and where can this be found? Connection is closed by server after one request has been processed. HTTP version 1.

When are non-persistent connections becoming inefficient? When Web documents contain embedded content such as images, multi-media content, etc.

How could this be overcome? Establish sever connections simultaneously to achieve parallelism with requests. But this will impact the load on the server, and may not be effective in slow networks.

Communication (2)

Example (b): Using persistent connections in HTTP.



Interactive slide

What is a persistent connection, and where can this be found?

What is realized through persistent connections?

When are persistent connections becoming inefficient?

Interactive slide

What is a persistent connection, and where can this be found? Connection stays open. Multiple requests can be made through a single connection. Supported by HTTP version 1.1 and greater.

What is realized through persistent connections? Pipelining of requests through the same connection.

When are persistent connections becoming inefficient? When servers are not designed to handle compound requests efficiently. Client make fast successive requests cause increased server load. Increased vulnerability to denial of service (DOS) attacks.

HTTP Methods

Request operations (primitives) supported by HTTP:

Operation	Description
Head	Request to return the header of a document
Get	Request to return a document to the client
Put	Request to store a document
Post	Provide data that are to be added to a document (collection)
Delete	Request to delete a document

Interactive slide

What information can be expected to be received in response to operation Head?

What is happening when trying to issue a Delete, Push, or Post request on a protected item?

Interactive slide

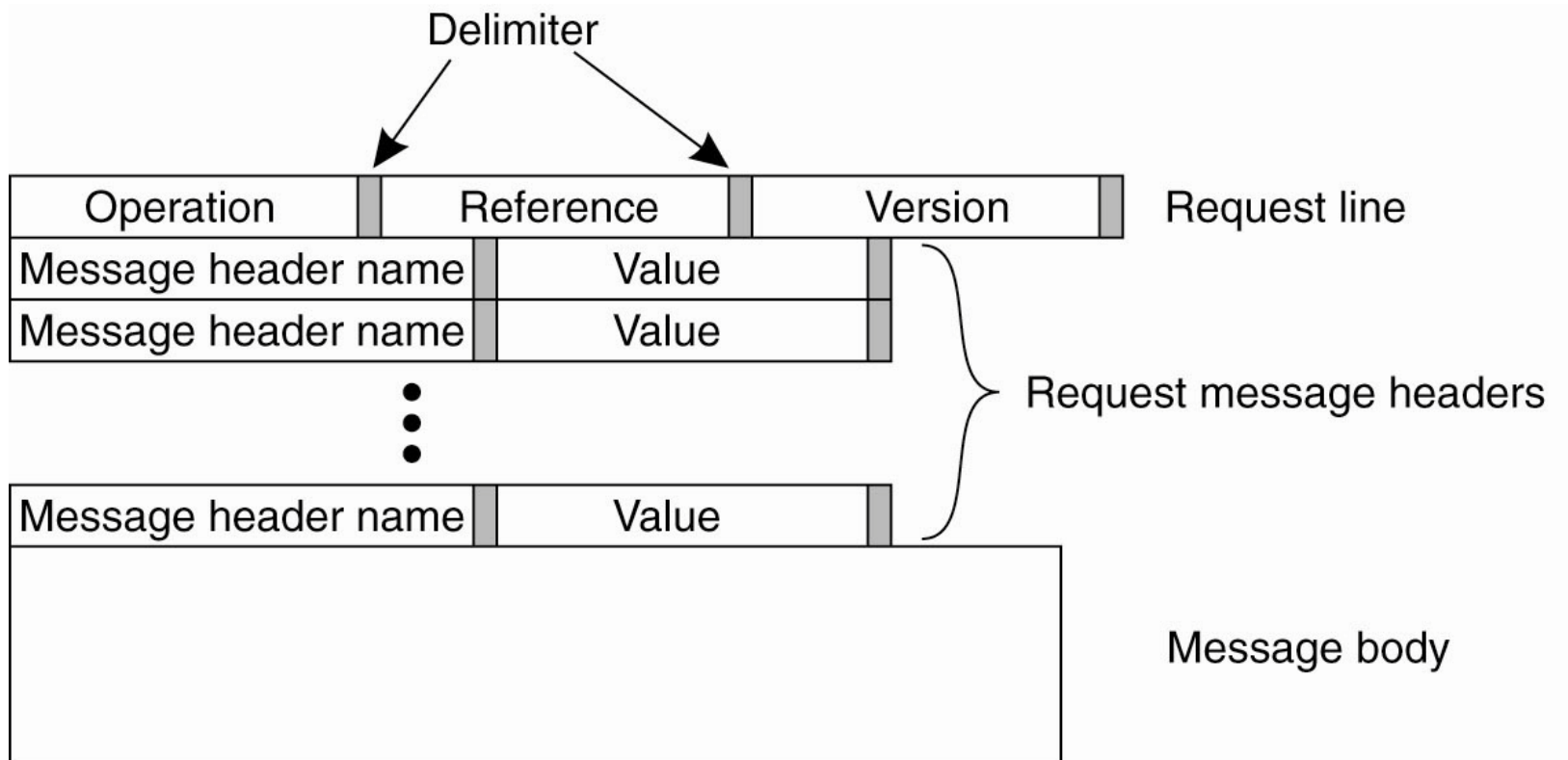
What information can be expected to be received in response to operation Head?

An error code (if document does not exist, or if insufficient access rights), date of last modification (to identify whether client's cached version is up to date).

What is happening when trying to issue a Delete, Push, or Post request on a protected item? A response containing an error code such as code 403 (Forbidden)

HTTP Messages (1)

With HTTP all request methods result in messages of the same form:



(a)

Interactive slide

Request Line optional: yes/no?

Message Headers optional: yes/no?

Message Body optional: yes/no?

What content can be found in the following message fields:

Operation:

Reference:

Version:

Interactive slide

Request Line optional: **no**

Message Headers optional: **yes**

Message Body optional: **yes**

What content can be found in the following message fields:

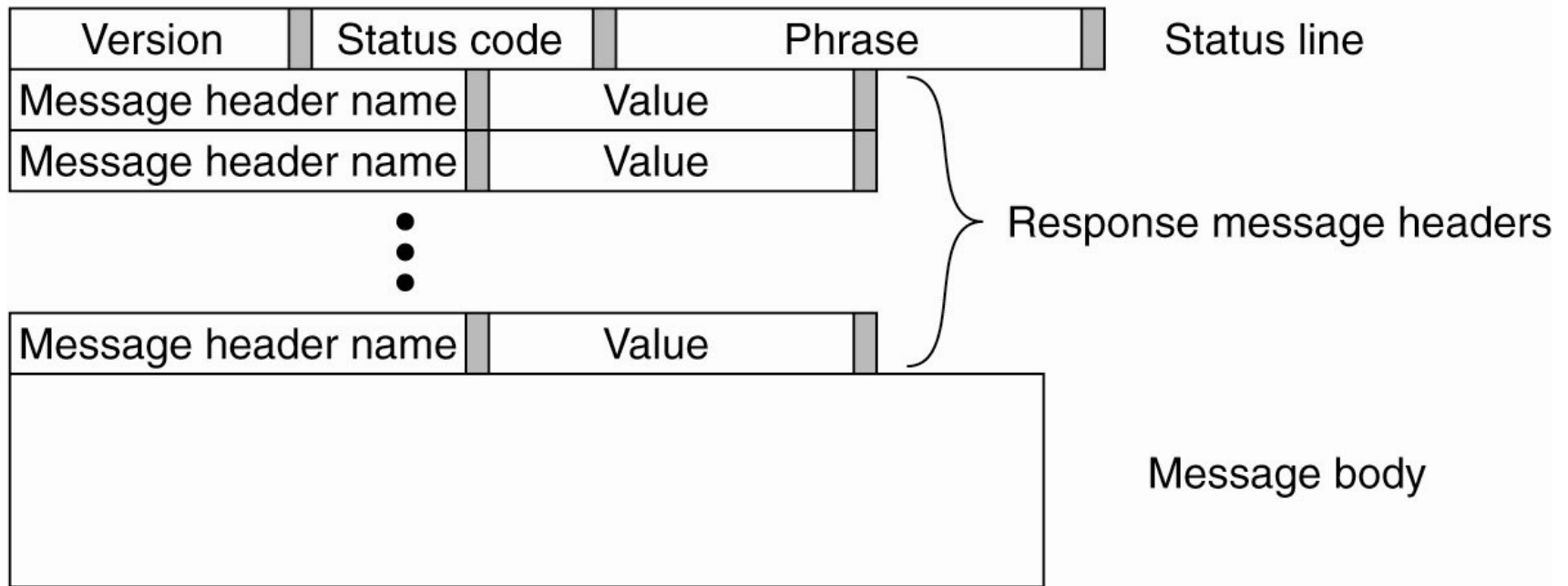
Operation: Operation to carry out (i.e. GET, PUT, etc)

Reference: Document to be affected by the operation.

Version: HTTP version the client is expecting.

HTTP Messages (2)

With HTTP all response messages are expected to be in the form as shown here:



(b)

Interactive slide

What content can be found in the following message fields:

Version:

Status code and phrase:

Message body:

Interactive slide

What content can be found in the following message fields:

Version: HTTP version of the message.

Status code and phrase: a 3-digit code and a brief explanation of the meaning (i.e. 404 (Not Found), or 405 (Method not allowed)).

Message body: Is an optional field which can contain data of a file (i.e. the content of a http file if a http file was requested).

HTTP Messages (3)

Some valid values for HTTP message header names.

Header	Source	Contents
Accept	Client	The type of documents the client can handle
Accept-Charset	Client	The character sets are acceptable for the client
Accept-Encoding	Client	The document encodings the client can handle
Accept-Language	Client	The natural language the client can handle
Authorization	Client	A list of the client's credentials
WWW-Authenticate	Server	Security challenge the client should respond to
Date	Both	Date and time the message was sent
ETag	Server	The tags associated with the returned document
Expires	Server	The time for how long the response remains valid
From	Client	The client's e-mail address
Host	Client	The DNS name of the document's server

Interactive slide

Explain Accept-Charset and give one possible value:

Explain Accept-Encoding and give one possible value:

Explain Expires:

Interactive slide

Explain Accept-Charset and give one possible value: This accounts for (human) languages which may differ in the way they are encoded on a computer, a charset identifier is a unique reference to a specific encoding of a language. Example: ISO8859 refers to a standard western charset, or GB2312 refers to the simplified Chinese charset.

Explain Accept-Encoding and give one possible value: The type of encoding of a document (i.e. plain text, gzip, jpeg, encryption methods, etc.)

Explain Expires: Explicitly tells a client for how long the content of a response remains valid.

HTTP Messages (4)

Some more HTTP message headers:

Header	Source	Contents
If-Match	Client	The tags the document should have
If-None-Match	Client	The tags the document should not have
If-Modified-Since	Client	Tells the server to return a document only if it has been modified since the specified time
If-Unmodified-Since	Client	Tells the server to return a document only if it has not been modified since the specified time
Last-Modified	Server	The time the returned document was last modified
Location	Server	A document reference to which the client should redirect its request
Referer	Client	Refers to client's most recently requested document
Upgrade	Both	The application protocol the sender wants to switch to
Warning	Both	Information about the status of the data in the message

Interactive slide

Explain difference between If-Modified-Since and Last-Modified:

Explain where the HTTP message “Upgrade” is useful:

Interactive slide

Explain difference between If-Modified-Since and Last-Modified: A client requests a document and expects the document to be returned only if the document has been modified since a given date and time.

Explain where the HTTP message “Upgrade” is useful: To negotiate the use of the latest version of HTTP supported by both sender and receiver. Initial request may use a low version number to ensure that a server does not respond with a error message.

Naming (1)

The common structure for URLs can allow to access same resource using different “names”. Example: (a) Using only a DNS name. (b) Combining a DNS name with a port number. (c) Combining an IP address with a port number.

Scheme	Host name	Pathname
--------	-----------	----------

http :// www.cs.vu.nl /home/steen/mbox

(a)

Scheme	Host name	Port	Pathname
--------	-----------	------	----------

http :// www.cs.vu.nl : 80 /home/steen/mbox

(b)

Scheme	Host name	Port	Pathname
--------	-----------	------	----------

http :// 130.37.24.11 : 80 /home/steen/mbox

(c)

Interactive slide

Port and Pathname in the naming scheme of HTTP are optional. What are their default values?

Which of the following hostnames are valid?

122.65.132.11.1.97

My_server

300.320.1.32

Interactive slide

Port and Pathname in the naming scheme of HTTP are optional. What are their default values? 80, and {‘/’, “index.htm”, “default.htm”, “index.html”, any path}. Can be configured freely on the server.

Which of the following hostnames are valid?

122.65.132.11.1.97

My_server

300.320.1.32

All three can be valid depending on the scheme used:

Naming (2)

Schemes are defined by means of URIs. Examples of URIs:

Name	Used for	Example
http	HTTP	http://www.cs.vu.nl:80/globe
mailto	E-mail	mailto:steen@cs.vu.nl
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:/edu/book/work/chp/11/11
data	Inline data	data:text/plain;charset=iso-8859-7,%e1%e2%e3
telnet	Remote login	telnet://flits.cs.vu.nl
tel	Telephone	tel:+31201234567
modem	Modem	modem:+31201234567;type=v32

Interactive slide

Definition of URL:

Definition of URI:

How does a web-browser interpret the various URIs?

Interactive slide

Definition of URL: Uniform Resource Locator

Definition of URI: Uniform Resource Identifiers

How does a web-browser interpret the various URIs?

- Inbuilt functions.
- Plugins (Code migration).
- Calls to system library functions.
- Calls to external applications.
- User feedback.
- Printing of a (error) message.

Interactive slide

Do you agree: Client side caching is not required if proxy caches are used.

Do you agree: Server side caching is not required if proxy caches are used.

How to reduce the risk that proxy caches contain out of date information?

Interactive slide

Do you agree: Client side caching is not required if proxy caches are used.

Answer depends on reasoning.

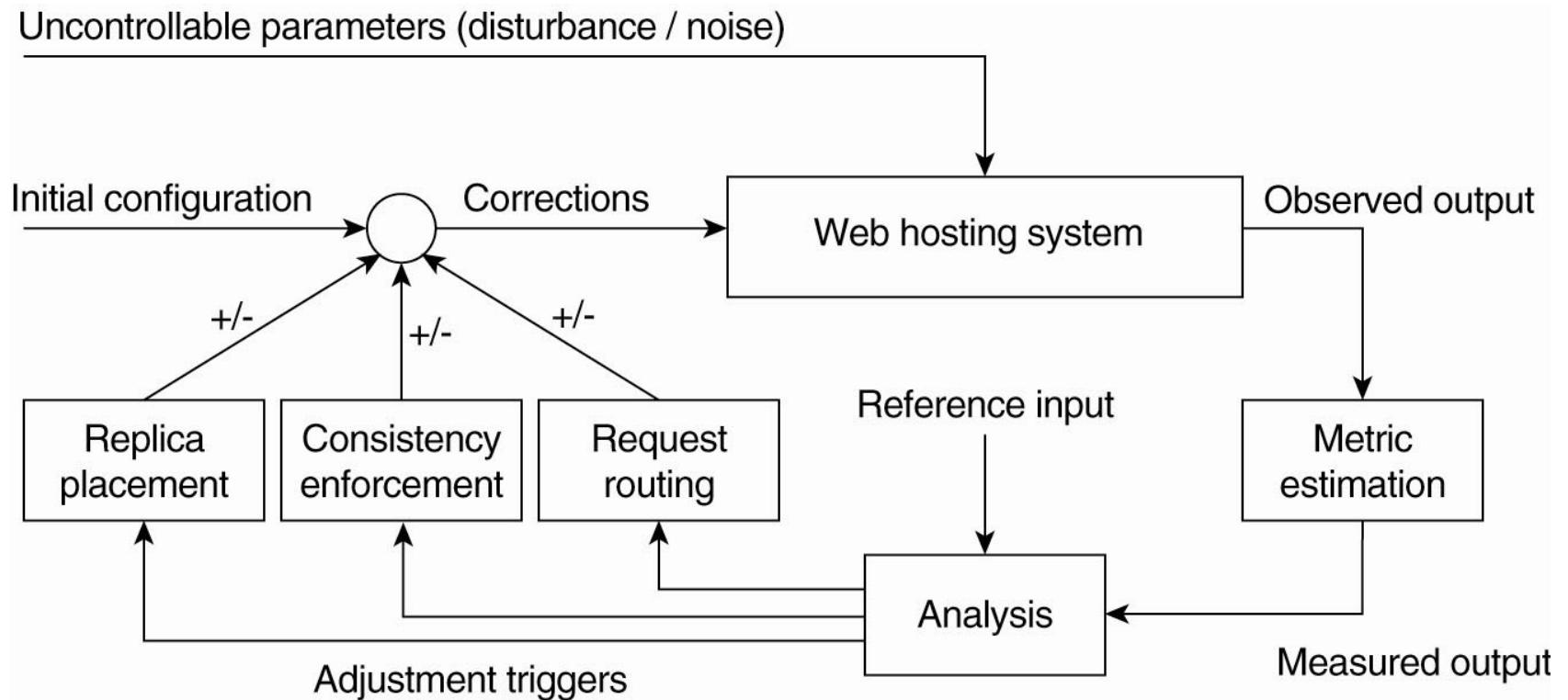
Do you agree: Server side caching is not required if proxy caches are used.

Answer depends on reasoning.

How to reduce the risk that proxy caches contain out of date information? Proxy to take a pro-active role by querying server for updates, or discard information which may be out of date, or request from server to inform proxy of relevant updates.

Replication for Web Hosting Systems

The general organization of a CDN as a feedback-control system :



Interactive slide

Definition of CDN:

Name advantages of a feedback control system in a CDN: .

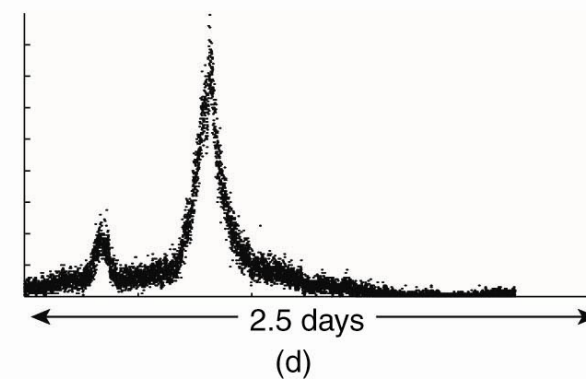
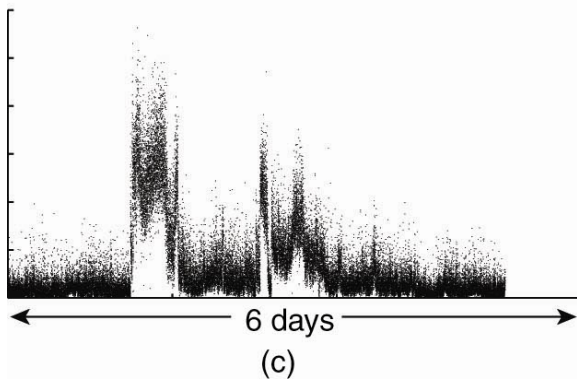
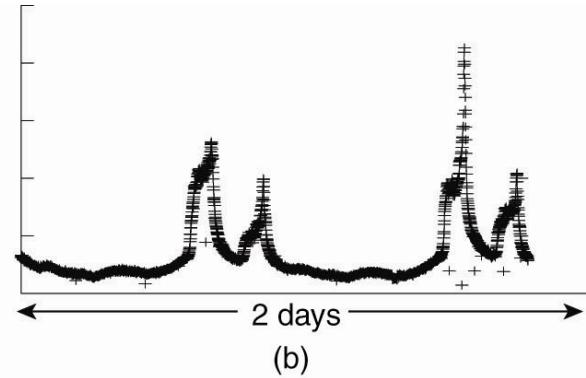
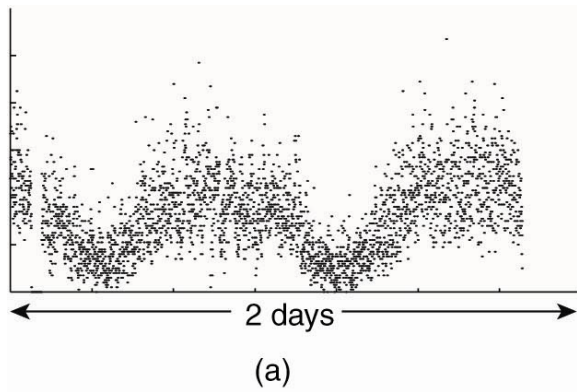
Interactive slide

Definition of CDN: Content delivery networks

Name advantages of a feedback control system in a CDN: Allows adjustments to daily changes in system load, can prevent DOS on Web server, mask failure of a Web server by relaying requests to non-faulty nodes, strategic distribution of Web content, automation of information distribution, and more.

Adaptation Triggering

One normal and three different access patterns reflecting flash-crowd behavior.



Interactive slide

Explain flash crowd:

Explain different types of a flash crowd, and how these can be predicted.

Interactive slide

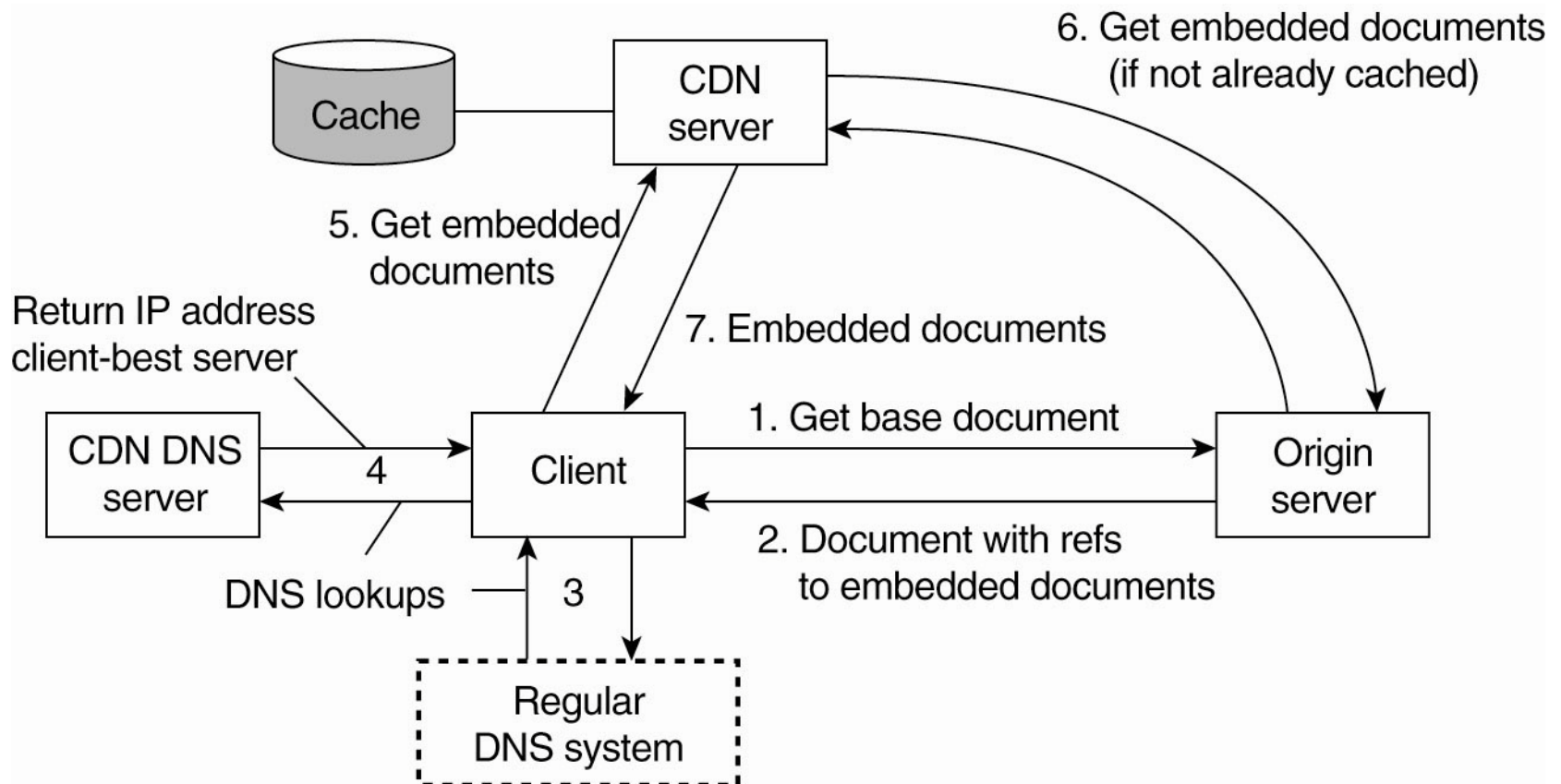
Explain flash crowd: Is a sudden burst in requests. It can be the result of a DOS attack.

Explain different types of a flash crowd, and how these can be predicted.

- 1.) Increased requests due to predictable events (i.e requests that occur at “busy” times of a day, are the result of information seeking behavior of the users of the system). Predict through analysis of long term access patterns.
- 2.) Increased requests due to unpredictable events (large number of dummy requests which exist for no reason other than to do harm). Predict through analysis of short term access patterns.

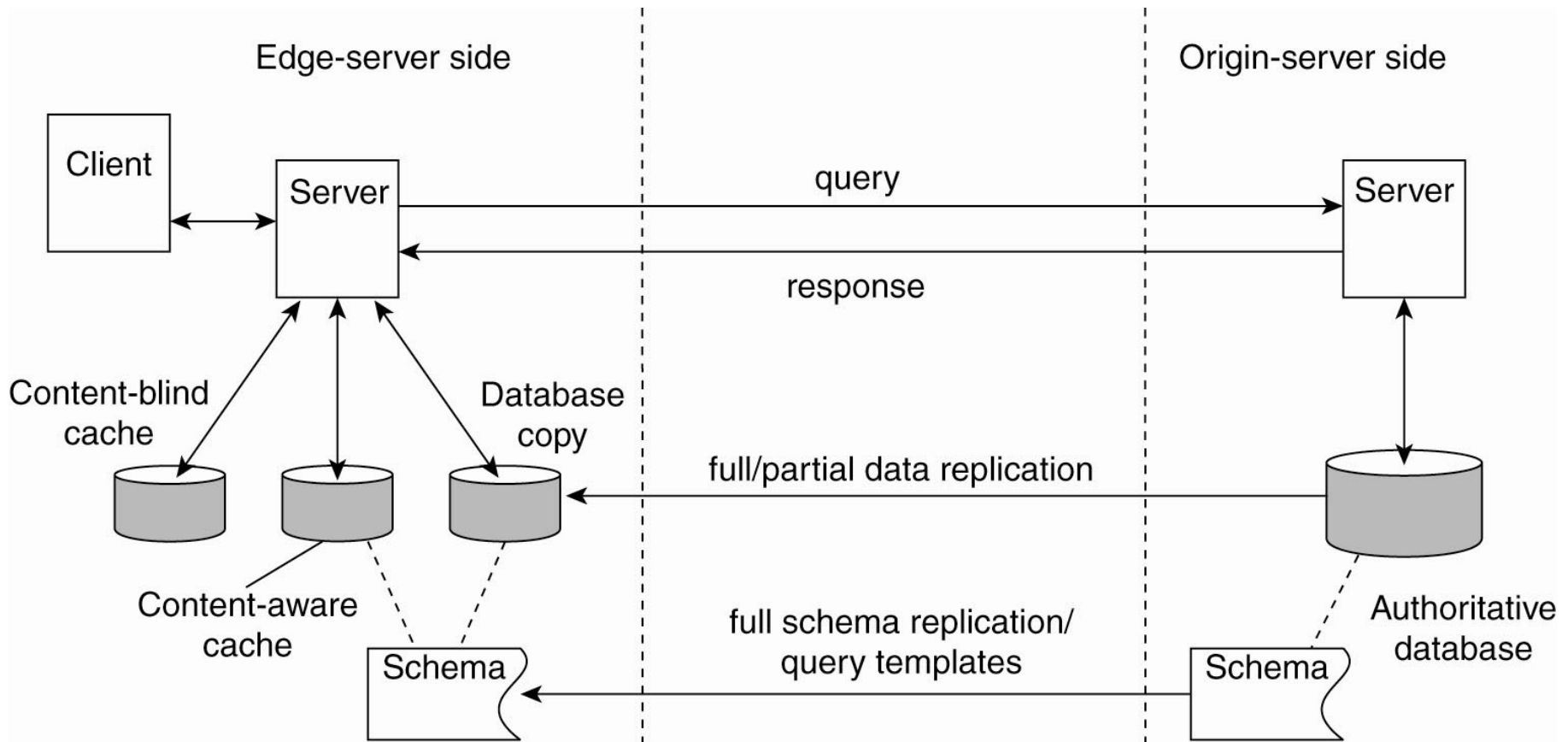
Adjustment Measures

The principal working of the Akamai CDN.



Replication of Web Applications

Alternatives for caching and replication with Web applications.



Interactive slide

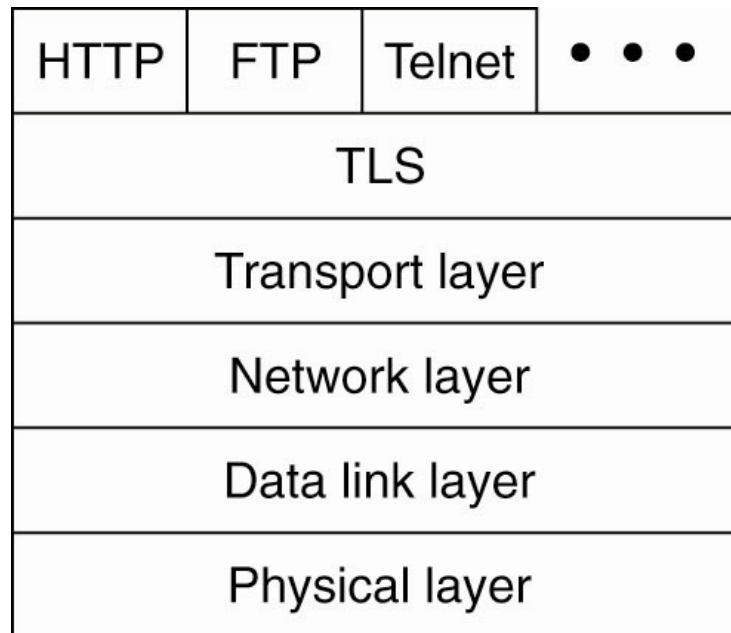
Explain the three different caches of the CDN:

1. Database copy:
2. Content aware cache:
3. Content-blind cache:

What strategies would you use if all CDN servers are on high load?

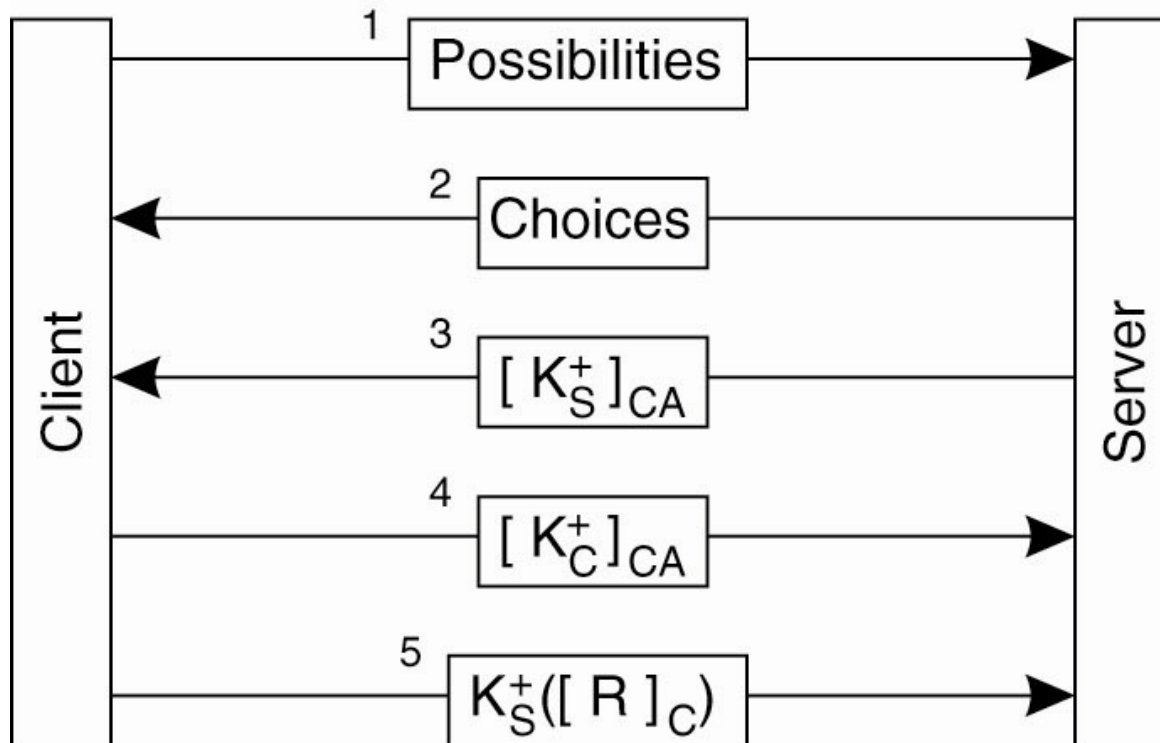
Security (1)

The open nature of the Internet requires a security architecture that protects clients and servers. The predominant approach is through Secure Socket Layer (SSL) realized as Transport Layer Security (TLS) below the application layer in the Internet protocol stack as illustrated here:



Security (2)

TLS with mutual authentication.



Summary

Web based system design:

- Architectures: Multi-tiered, clusters
- Processes: Client and server processes
- Naming: Structured naming in HTTP
- Fault tolerance: Client side caching, server replication
- Synchronization: i.e. Distributed Authoring use locking (WebDAV, not covered)
- Consistency and Replication: Caching, feedback control
- Security: Mainly Encryption and authentication